File No. ____________

Commenced From ____________

Closed on ____________

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

Shobhavana Campus, Mijar, Moodbidri, D.K – 574225

Phone: 08258-262725, Fax: 08258-262726

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

COURSE FILE for the Academic Year 2022-2023 (Even Semester)

Course Name : Java for Mobile Applications.

Course Code : 18AI63

Faculty Name : SHRIKANTH N.G.

Designation : Sr. Assistant Professor.

Course File Checklist

Sl.No.	Contents
1.	Cover Page
2.	College Calendar
3.	Time Table(Class + Personal)
4.	Course Content (Syllabus)
5.	Course Outcomes & Mapping Matrix
6.	Lesson Plan/Execution
7.	Attendance Register
8.	Internal Test-I Question Paper & Scheme
9.	Internal Test-II Question Paper & Scheme
10.	Internal Test-III Question Paper & Scheme
11.	All Assignment Questions and Scheme(Rubrics)
12.	All Assignment Marks
13.	Final Internal Marks & Attendance Statement(As per Attendance Register)
14.	University Question Papers
15.	Notes/PPTs/Handouts
16.	Student Feedback Assessment Form(Coursewise and Course End Survey)
17.	University Result Analysis Statement
18.	Attainment of the Course (As per CO/PO-PSO Mapping Matrix's)
19.	List of Weak students and Bright students
20.	Sample Laboratory Observation and Record Book(only for Lab Courses)
21.	Sample Laboratory Internal Test Answer Sheets(only for Lab Courses)
22.	Project/ Seminar Report Formats and Regulations
23.	Other documents (if any)

College: Alva's Institute of Engineering and Technology

VISION

"Transformative education by pursuing excellence in Engineering and Management through enhancing skills to meet the evolving needs of the community"

MISSION

1. To bestow quality technical education to imbibe knowledge, creativity and ethos to students community.
2. To inculcate the best engineering practices through transformative education.
3. To develop a knowledgeable individual for a dynamic industrial scenario.
4. To inculcate research, entrepreneurial skills and human values in order to cater the needs of the society.

Department: Artificial Intelligence and Machine Learning

VISION

Foster competent professionals by instilling knowledge and skills in the Artificial Intelligence and Machine Learning realm to cater needs of industry and community.

MISSION

- To strengthen the assimilation process of concepts in AI & ML through experiential learning.
- To create a better Academia-Industry liaison by means of skill enhanced training.
- To develop a support system for Research and Development for broader application in AIML domain.
- To promote Entrepreneurial culture through interaction with collaborative knowledge partners.

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(A Unit of Alva's Education Foundation)

Shobhavana Campus, Mijar-574225, Moodbidri, D.K

Phone: 08258-262725, Fax: 08258-262726

Affiliated to VTU Belagavi and Approved by AICTE, New Delhi, Recognized by Govt. of Karnataka

(Accredited by NAAC with A+ Grade)

VISION

"Transformative education by pursuing excellence in Engineering and Management through enhancing skills to meet the evolving needs of the community"

MISSION

- To bestow quality technical education to imbibe knowledge, creativity and ethos to students community.
- To inculcate the best engineering practices through transformative education.
- To develop a knowledgeable individual for a dynamic industrial scenario.
- To inculcate research, entrepreneurial skills and human values in order to cater the needs of the society.

Week	Month	Days							Activities
		Mon	Tue	Wed	Thu	Fri	Sat	Sun	
1	FEB	13	14	15	16	17	18	19	13 : Commencement of VIII Semester
2		20	21	22	23	24	25	26	
3		27	28						
4	MAR			1	2	3	4	5	20 : Commencement of VI Semester 22nd : Chandramana Ugadi 27 – 31 : Technical Talk/Club and Social Activity 30 - 31 : 1st IA for VIII Semester
5		6	7	8	9	10	11	12	
6		13	14	15	16	17	18	19	
7		20	21	22	23	24	25	26	
8		27	28	29	30	31			
9	APR						1	2	3 : Mahaveera Jayanthi 7: Good Friday 14: Dr B.R. Ambedkar Jayanti 22: Khutha-e-Ramzan 20-21 – Student Mentoring 26 : College Level Project Exhibition 27-28 : 2nd IA for VIII Semester 24- 29 Technical Talk/Club / Social Activity
10		3	4	5	6	7	8	9	
11		10	11	12	13	14	15	16	
12		17	18	19	20	21	22	23	
13		24	25	26	27	28	29	30	
14	MAY	1	2	3	4	5	6	7	1 : Labor day 6 : Sports Day 8-9 : 3rd IA for VIII Semester 13 : Last Working Day of VIII Semester 17 : Commencement of IV Semester 20 : Traditional Day. 22 : College Day Celebration 25 : Commencement of II Semester 26 : Farewell Function to Final Years 22-23 : Student Mentoring 25 - 27 : 1st IA for VI Semester 29-31 : Technical Talk/Club / Social Activity
15		8	9	10	11	12	13	14	
16		15	16	17	18	19	20	21	
17		22	23	24	25	26	27	28	
18		29	30	31					
19	JUN				1	2	3	4	16 To 19 : 2nd IA for VI Semester 26-27 : Student Mentoring 20 - 24 : Technical Talk/Club / Social Activity 28 , 30 & 1st July : 1st IA for IV Semester 30/Jun to 4/July : 1st IA for II Semester 29- Bakrid
20		5	6	7	8	9	10	11	
21		12	13	14	15	16	17	18	
22		19	20	21	22	23	24	25	
23		26	27	28	29	30			
24	JULY						1	2	1-4 : 1st IA for II Semester 5 – 7 : 3rd IA for VI Semester 10 : Last Working Day of VI Semester 17 - 22 : Technical Talk/Club / Social Activity 24-25 : Student Mentoring 29 : Last Day of Moharamm
25		3	4	5	6	7	8	9	
26		10	11	12	13	14	15	16	
27		17	18	19	20	21	22	23	
28		24	25	26	27	28	29	30	
29		31							
30	AUG		1	2	3	4	5	6	4 To 8 : 2nd IA for II Semester 4 To 7 : 2nd IA for IV Semester 15 : Independence Day 24-25 : Student Mentoring 28 - 31 : Technical Talk/Club / Social Activity
31		7	8	9	10	11	12	13	
32		14	15	16	17	18	19	20	
33		21	22	23	24	25	26	27	
34		28	29	30	31				
35	SEP					1	2	3	1 To 5 : 3rd IA for II Semester 8 To 11 : 3rd IA for IV Semester 9 : Last Working Day of II Semester 16 : Last Working Day of IV Semester
36		4	5	6	7	8	9	10	
37		11	12	13	15	16	17	18	

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

Shobhavana Campus, Mijar, Moodbidri, D.K – 574225
Phone: 08258-262725, Fax: 08258-262726

Academic Time Table W.E.F. 20/03/2023

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Academic Year	Scheme	Semester	Section	Room No	Class Coordinator
2022-23	2018	VI	A	201	Mr. Shrikanth N G

Time Day	9.00 To 9.55	9.55 To 10.50	10.50 To 11.00	11.00 To 12 Noon	12 Noon To 1:00	1:00 To 2:00	2:00 To 3:00	3:00 To 4:00	4:00 To 5.00
MON	ML	JMA	TEA BREAK	PDC		LUNCH BREAK	DS (SBC)	DIP	OEC
TUE	JMA	ML		DS (SBC)	DIP		Digital Image Processing Laboratory with Mini Project		OEC
WED	DIP	DS (SBC)		PDC			ML	JMA	OEC
THU	DS (RG)	DIP		JMA	ML		Machine Learning Laboratory		OEC
FRI	DS (RG)	ML		DIP	JMA		Mobile Application Development Laboratory		OEC
SAT	DIP	JMA		ML	SAP				

Allocation of Courses

Course Code & Course Title			Faculty In Charge	Faculty Code
18AI61	ML	Machine Learning [PCC]	Mr. Apurba Chakraborty	AC
18AI62	DIP	Digital Image Processing [PCC]	Mr. Kiran Raj K M	KR
18AI63	JMA	Java for Mobile Applications [PCC]	Mr. Shrikanth N G	SNG
18AI644	DS	Foundation for Data Science [PEC]	Ms. Soundarya B C Dr. Ramesh G	SBC RG
18ME651	NCES	Non – Conventional Energy Sources [OEC] R.No. 302 (ISE) / 302 (ECE)	Mr. Hemanth Suvarna	HS
18ME653	SCM	Supply Chain Management [OEC] R. No. 201 (AIML) / 303 (ECE)	Mr. Deepak Kothari	DK
18CV651	RSGIS	Remote Sensing & GIS [OEC] R. No. 503 (CV)	Dr. H G Umeschandra	HGU
18CV653	OHS	Occupational Health & Safety [OEC] R. No. 312 (CSE) / 402 (ME)	Ms. Anusha B Rao	ABR
18AIL66	ML Lab	Machine Learning Laboratory	Mr. Apurba Chakraborty Prof. Harish Kunder	AC HK
18AIMP67	DIP Lab	Digital Image Processing Laboratory with Mini Project [MP]	Mr. Kiran Raj K M Dr. Ramesh G	KR RG
18AIL68	MAD Lab	Mobile Application Development Laboratory	Mr. Shrikanth N G Mrs. Rashmi Suvarna	SNG RS
PDC		Personality Development Courses	Mr. Deviprasad Shetty Airodynamiks	
SAP		Seminar / Aptitude Test / Project	Prof. Harish Kunder	

PCC: Professional Core Course, **PEC:** Professional Elective Course, **OEC:** Open Elective Course, **MP:** Mini Project

Time Table Coordinator
16/03/2023

Head of the Department
Dept. of Artificial Intelligence & Machine Learning
Alva's Institute of Engineering and Technology
Shobhavana Campus, Mijar
Moodubidire 574 225, D.K. Karnataka, India

Principal

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

Shobhavana Campus, Mijar, Moodbidri, D.K – 574225
Phone: 08258-262725, Fax: 08258-262726

Academic Time Table W.E.F. 20/03/2023

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Academic Year	Scheme	Semester	Section	Room No	Class Coordinator
2022-23	2018	VI	A	201	Mr. Shrikanth N G

Time / Day	9.00 To 9.55	9.55 To 10.50	10.50 To 11.00	11.00 To 12 Noon	12 Noon To 1:00	1:00 To 2:00	2:00 To 3:00	3:00 To 4:00	4:00 To 5.00
MON	ML	JMA	TEA BREAK	PDC		LUNCH BREAK	DS	DIP	OEC
TUE	JMA	ML		DS	DIP		Digital Image Processing Laboratory with Mini Project		OEC
WED	DIP	DS		PDC			ML	JMA	OEC
THU	DS	DIP		JMA	ML		Machine Learning Laboratory		OEC
FRI	DS	ML		DIP	JMA		Mobile Application Development Laboratory		OEC
SAT	DIP	JMA		ML	SAP				

Allocation of Courses

Course Code & Course Title			Faculty In Charge	Faculty Code
8AI61	ML	Machine Learning [PCC]	Mr. Apurba Chakraborty	AC
8AI62	DIP	Digital Image Processing [PCC]	Mr. Kiran Raj K M	KR
8AI63	JMA	Java for Mobile Applications [PCC]	Mr. Shrikanth N G	SNG
8AI644	DS	Foundation for Data Science [PEC]	Ms. Soundarya B C	SBC
8ME651	NCES	Non – Conventional Energy Sources [OEC] R.No. 302 (ISE) / 302 (ECE)	Mr. Hemanth Suvarna	HS
8ME653	SCM	Supply Chain Management [OEC] R. No. 201 (AIML) / 303 (ECE)	Mr. Deepak Kothari	DK
8[illegible]51	RSGIS	Remote Sensing & GIS [OEC] R. No. 503 (CV)	Dr. H G Umeschandra	HGU
8CV653	OHS	Occupational Health & Safety [OEC] R. No. 312 (CSE) / 402 (ME)	Ms. Anusha B Rao	ABR
8AIL66	ML Lab	Machine Learning Laboratory	Mr. Apurba Chakraborty	AC
8AIMP67	DIP Lab	Digital Image Processing Laboratory with Mini Project [MP]	Mr. Kiran Raj K M	KR
8AIL68	MAD Lab	Mobile Application Development Laboratory	Mr. Shrikanth N G	SNG
PDC		Personality Development Courses	Mr. Deviprasad Shetty Airodynamiks	
SAP		Seminar / Aptitude Test / Project	Prof. Harish Kunder	

PCC: Professional Core Course, **PEC:** Professional Elective Course, **OEC:** Open Elective Course, **MP:** Mini Project

17/03/2023.
Time Table Coordinator

17/03/2023
Head of the Department

Head of the Department
Dept. of Artificial Intelligence & Machine Learning
Alva's Institute of Engineering and Technology
Shobhavana Campus, Mijar
Moodubidire 574 225, D.K. Karnataka, India

Principal

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(A Unit of Alva's Education Foundation)
Shobhavana Campus, Mijar, Moodbidri, D.K – 574225
(Accredited by NAAC with A+ Grade)
Affiliated to VTU Belagavi, Approved by AICTE, New

Individual Faculty Time Table with effect from 20/03/2023
DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Academic Year	2022-23	Faculty Name	Mr. Shrikanth N G (SNG)
Semester	EVEN	Designation	Senior Assistant Professor

Time / Day	9.00 To 9.50	9.50 To 10.40	10.40 To 11.00	11.00 To 11.50	11.50 To 12.40	12.40 To 1.40	1.40 To 2.30	2.30 To 3.20	3.30 To 5.00
MON	PP LAB AIML	JMA	TEA BREAK	PP LAB AIML		LUNCH BREAK	OS		
TUE	JMA				OS				
WED		OS						JMA	
THU				JMA	R Pgming AIML			OS	
FRI	OS				JMA		MAD LAB		
SAT		JMA							

UNITS:	Theory:22	LAB: 05	Others: 01	TOTAL UNITS: 28

Allocation of Courses (Courses with Course Code)	
21CS44	Operating Systems
18AI63	Java for Mobile Applications [PCC]
18AIL68	Mobile Application Development Laboratory
21CSL46	Python Programming Laboratory
21CSL483	R – Programming [AEC]
	Mentoring (2Hrs / Week)

Responsibilities
Training & Placement Coordinator
Forum Coordinator
Students Consultancy Projects, Industry Visits, Industry Interaction, MOU In charge
NBA Coordinator
Seminar Coordinator, Class Coordinator

Head of the Department   Dean Academics   PRINCIPAL

KHP 20/03/2023

Head of the Department
pt. of Artificial Intelligence & Machine Learning
lva's Institute of Engineering and Technology
Shobhavana Campus, Mijar

JAVA FOR MOBILE APPLICATIONS
(Effective from the academic year 2018 -2019)
SEMESTER – VI

Subject Code	18AI63	**CIE Marks**	40
Number of Contact Hours/Week	3:2:0	**SEE Marks**	60
Total Number of Contact Hours	50	**Exam Hours**	3 Hrs

CREDITS –4

Course Learning Objectives: This course will enable students to:

- To have an insight into enumerations and collection frameworks for storing and processing data.
- To understand the architecture and components of android application.
- To design interactive user interface.
- To work with SQLite database

Module 1	**Contact Hours**
Enumerations, Autoboxing and Annotations(metadata): Enumerations, Enumeration fundamentals, the values () and valueOf() Methods, java enumerations are class types, enumerations Inherits Enum, example, type wrappers, Autoboxing, Autoboxing and Methods, Autoboxing/Unboxing occurs in Expressions, Autoboxing/Unboxing, Boolean and character values, Autoboxing/Unboxing helps prevent errors, A word of Warning. Annotations, Annotation basics, specifying retention policy, Obtaining Annotations at run time by use of reflection, Annotated element Interface, Using Default values, Marker Annotations, Single Member annotations, Built-In annotations. **RBT: L2, L3**	10
Module 2	
The collections and Framework: Collections Overview, Recent Changes to Collections, The Collection Interfaces, The Collection Classes, accessing a collection Via an Iterator, Storing User Defined Classes in Collections, The Random Access Interface, Working with Maps, Comparators, The Collection Algorithms, Why Generic Collections? The legacy Classes and Interfaces, Parting Thoughts on Collections **RBT: L1, L2**	10
Module 3	
String Handling: The String Constructors, String Length, Special String Operations, String Literals, String Concatenation, String Concatenation with Other Data Types, String Conversion and toString( ) Character Extraction, charAt( ), getChars( ), getBytes( ) toCharArray(), String Comparison, equals( ) and equalsIgnoreCase( ), regionMatches( ) startsWith( ) and endsWith( ), equals( ) Versus ==, compareTo( ) Searching Strings, Modifying a String, substring( ), concat( ), replace( ), trim( ), Data Conversion Using valueOf( ), Changing the Case of Characters Within a String, Additional String Methods, StringBuffer, StringBuffer Constructors, length( ) and capacity( ), ensureCapacity( ), setLength( ), charAt( ) and setCharAt( ), getChars( ),append( ), insert( ), reverse( ), delete( ) and deleteCharAt( ), replace( ), substring( ), Additional StringBuffer Methods, StringBuilder **Text Book 1: Ch 15**	10
Module 4	
Getting Started with Android Programming: What is Android? Features of Android, Android Architecture, obtaining the required tools, launching your first android application **Activities, Fragments and Intents:** Understanding activities, linking activities using intents, fragments.**Text Book 3: Ch 1, 3**	10

RBT: L1, L2, L3	
Module 5	
Getting to know the Android User Interface: Views and ViewGroups, FrameLayout, LinearLayout, TableLayout, RelativeLayout, ScrollView **Designing User Interface with Views:** TextView view – Button, ImageButton, EditText, Checkbox, ToggleButton, RadioButton and RadioGroupViews. **Creating and using Databases:** Creating the DBAdapter Helper class, using the database programmatically. **Text Book 3: Ch 4.1, 5.1, 7.3** **RBT: L1, L2, L3**	10

Course Outcomes: The student will be able to:

- Interpret the need for advanced Java concepts like enumerations and collections in developing modular and efficient programs
- Understand various application components in android.
- Design efficient user interface using different layouts.
- Develop application with persistent data storage using SQLite

Question Paper Pattern:

- The question paper will have ten questions.
- Each full Question consisting of 20 marks
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

1.Herbert Schildt: JAVA the Complete Reference, 7th/9th Edition, Tata McGraw Hill, 2007.
2.Jim Keogh: J2EE-TheCompleteReference, McGraw Hill, 2007
3.J. F. DiMarzio, Beginning Android Programming with Android Studio, 4thEdition, 2017

Reference Books:

1. John Horton,Android Programming for Beginners, 1stEdition, 2015
2.Dawn Griffiths & David Griffiths, Head First Android Development, O'Reilly, 1stEdition, 2015

MOBILE APPLICATION DEVELOPMENT LABORATORY

(Effective from the academic year 2018 -2019)

SEMESTER – VI

Course Code	**18AIMP68**	**IA Marks**	40
Number of Contact Hours/Week	0:2:2	**Exam Marks**	60
Total Number of Contact Hours	3 Hours/Week	**Exam Hours**	03

CREDITS – 02

Course Learning Objectives: This course will enable students to:

- Learn and acquire the art of AndroidProgramming.
- Configure Android studio to run theapplications.
- Understand and implement Android's User interfacefunctions.
- Create, modify and query on SQlitedatabase.
- Inspect different methods of sharing data usingservices.

Descriptions (if any):

1. Installation procedure of the Android Studio/Java software must be demonstrated and carried out ingroups.

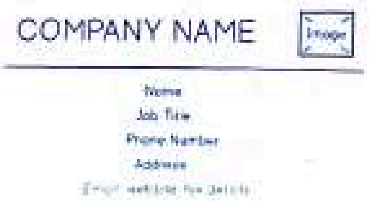
2. Students should use the latest version of Android Studio/Java/Kotlin to execute these programs. Diagrams given are for representational purpose only, students are expected to improvise on it.

3. Part B programs should be developed as an application and be demonstrated as a mini project in a group by adding extra features or the students can also develop their own application and demonstrate it as a mini project. (Projects/programs are not limited to the list given in Part B)

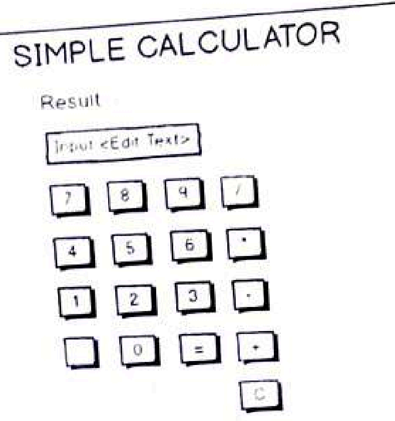
Programs List:

PART – A

1 Create an application to design aVisiting Card. The Visiting card should haveacompanylogoatthe top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address isto be displayed. Insert a horizontal line between the job title and the phone number.



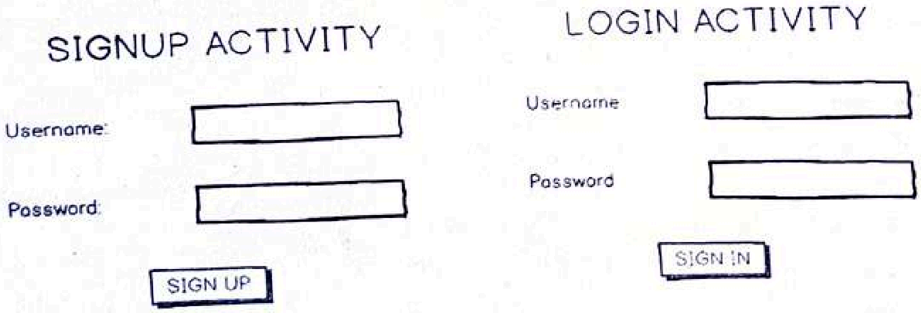
2 Develop an Android application usingcontrols like Button, TextView, EditText for designing a calculatorhaving basic functionality like Addition, Subtraction, Multiplication,andDivision.

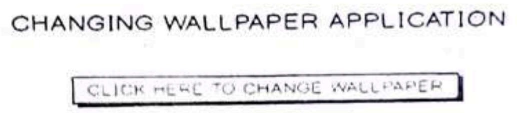
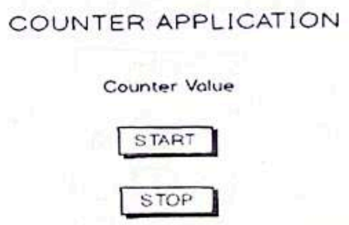
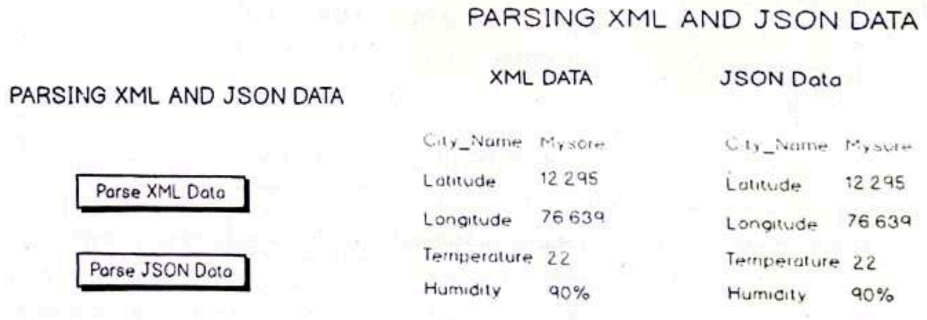


3 Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules:

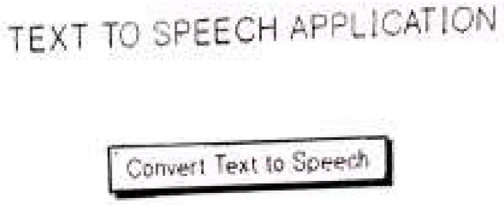
- Password should contain uppercase and lowercaseletters.
- Password should contain letters andnumbers.
- Password should contain specialcharacters.
- Minimum length of the password (the default value is8).

On successful **SIGN UP** proceed to the next Login activity. Here the user should **SIGN IN** using the Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity whichdisplays a message saying "Successful Login" or else display a toast message saying "Login Failed".The user is given only two attempts and after thatdisplay a toast message saying "Failed Login Attempts" and disable the SIGN IN button. Use Bundle to transfer information from one activity to another.

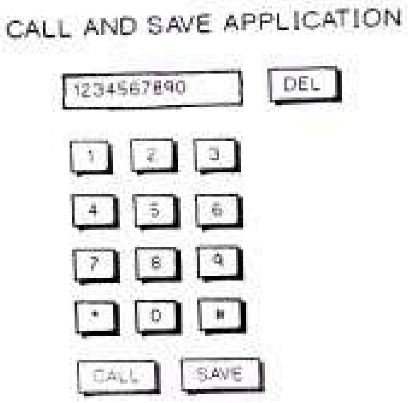


4	Develop an application to set an image as wallpaper. On click of a button, the wallpaper image should start to change randomly every 30 seconds.  CHANGING WALLPAPER APPLICATION CLICK HERE TO CHANGE WALLPAPER
5	Write a program to create an activity with two buttons START and STOP. On pressin g of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextViewcontrol. 
6	Create two files of XML and JSON type with values for City_Name, Latitude, Longitude, Temperature, and Humidity. Develop an application to create an activity with two buttons to parse the XML and JSON files which when clicked should display the data in their respective layouts side by side.  PARSING XML AND JSON DATA PARSING XML AND JSON DATA Parse XML Data Parse JSON Data XML DATA City_Name Mysore Latitude 12 295 Longitude 76 639 Temperature 22 Humidity 90% JSON Data City_Name Mysore Latitude 12 295 Longitude 76 639 Temperature 22 Humidity 90%

7 Develop a simple application with one Edit Text so that the user can write some text in it. Create a button called "Convert Text to Speech" that converts the user input text into voice.

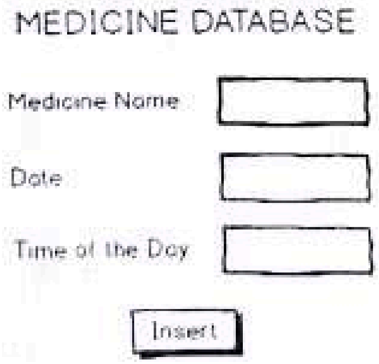


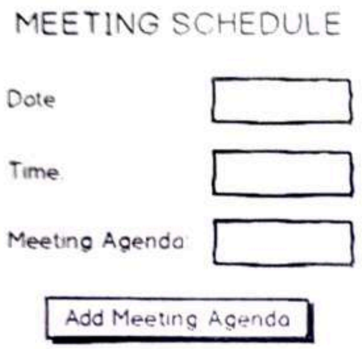
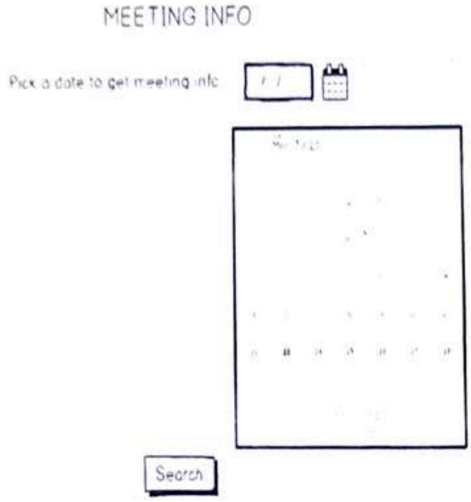
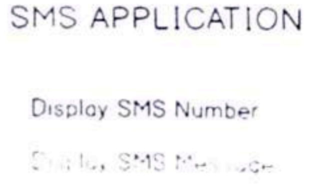
8 Create an activity like a phone dialer with CALL and SAVE buttons. On pressing the CALL button, it must call the phone number and on pressing the SAVE button it must save the number to the phonecontacts.

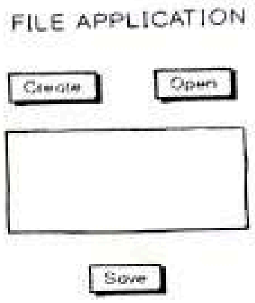
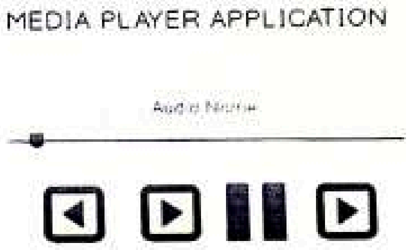
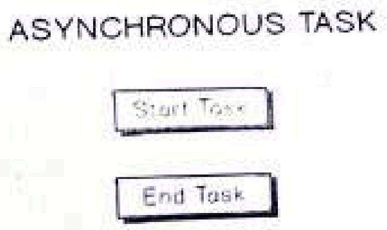


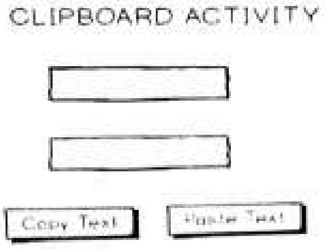
PART - B

1 Write a program to enter Medicine Name, Date and Time of the Day as input from the user and store it in the SQLite database. Input for Time of the Day should be either Morning or Afternoon or Evening or Night. Trigger an alarm based on the Date and Time of the Day and display the Medicine Name.



2	Develop a content provider application with an activity called "Meeting Schedule" which takes Date, Time and Meeting Agenda as input from the user and store this information into the SQLite database. Create another application with an activity called "Meeting Info" having DatePicker control, which on the selection of a date should display the Meeting Agenda information for that particular date, else it should display a toast message saying "No Meeting on this Date".  MEETING SCHEDULE Date Time Meeting Agenda Add Meeting Agenda 
3	Create an application to receive an incoming SMS which is notified to the user. On clicking this SMS notification, the message content and the number should be displayed on the screen. Use appropriate emulator control to send the SMS message to your application.  SMS APPLICATION Display SMS Number Display SMS Message
4	Write a program to create an activity having a Text box, and also Save, Open and Create buttons. The user has to write some text in the Text box. On pressing the Create button the text should be saved as a text file in MkSDcard. On subsequent changes to the text, the Save button should be pressed to store the latest content to the same file. On pressing the Open button, it should display the contents from the previously stored files in the Text box. If the user tries to save the contents in the Textbox to a file without creating it, then a toast message has to be displayed saying "First Create aFile".

	
5	Create an application to demonstrate a basic media player that allows the user to Forward, Backward, Play and Pause an audio. Also, make use of the indicator in the seek bar to move the audio forward or backward as required. 
6	Develop an application to demonstrate the use of Asynchronous tasks in android. The asynchronous task should implement the functionality of a simple moving banner. On pressing the **Start Task** button, the banner message should scroll from right to left. On pressing the **Stop Task** button, the banner message should stop. Let the banner message be "Demonstration of Asynchronous Task". 
7	Develop an application that makes use of the clipboard framework for copying and pasting of the text. The activity consists of two Edit Text controls and two Buttons to trigger the copy and paste functionality.



8 Create an AIDL service that calculates Car Loan EMI. The formula to calculate EMI is

$$E = P * (r(1+r)^n)/((1+r)^n-1)$$

where

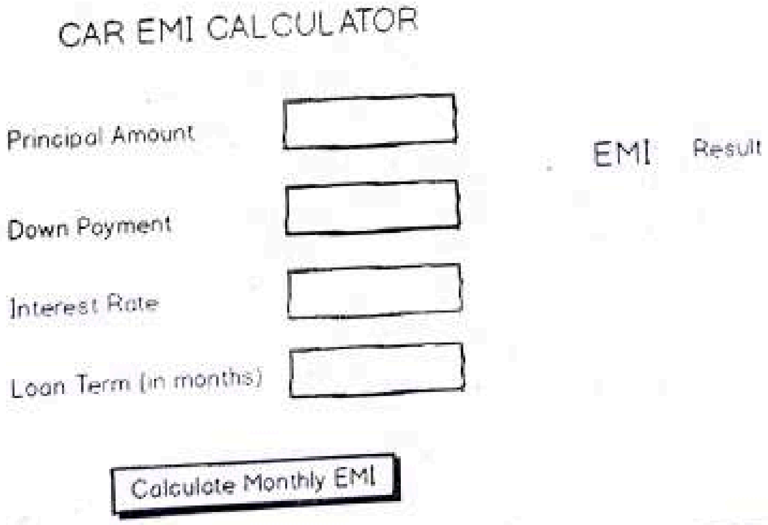
E = The EMI payable on the car loan amount

P = The Car loan Principal Amount

r = The interest rate value computed on a monthly basis

n = The loan tenure in the form of months

The down payment amount has to be deducted from the principal amount paid towards buying the Car. Develop an application that makes use of this AIDL service to calculate the EMI. This application should have four Edit Text to read the Principal Amount, Down Payment, Interest Rate, Loan Term (in months) and a button named as "Calculate Monthly EMI". On click of this button, the result should be shown in a Text View. Also, calculate the EMI by varying the Loan Term and Interest Rate values.



Laboratory Outcomes: After studying these laboratory programs, students will be able to

- Create, test and debug Android application by setting up Android developmentenvironment.
- Implement adaptive, responsive user interfaces that work across a wide range ofdevices.
- Infer long running tasks and background work in Androidapplications.
- Demonstrate methods in storing, sharing and retrieving data in Androidapplications.

- Infer the role of permissions and security for Androidapplications.

Procedure to Conduct Practical Examination

- **Experiment distribution**
  - **For laboratories having only one part:** Students are allowed to pick one experiment from the lot with equal opportunity.
  - **For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A with equal opportunity and in Part B demonstrate the Mini project.**
- **Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.**
- **Marks Distribution** *(Subjected to change in accoradance with university regulations)*
  - q) **For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+70+15 = 100 Marks**
  - r) **For laboratories having PART A and PART B**
    - i. **Part A – Procedure + Execution + Viva = 6 + 28 + 6 = 40 Marks**
    - ii. **Part B – Procedure + Execution + Viva = 9 + 42 + 9 = 60 Marks**

Text Books:

1. Google Developer Training. **"Android Developer Fundamentals Course – Concept Reference"**, Google Developer Training Team, 2017. https://www.gitbook.com/book/google-developer-training/android-developer-fundamentals-course-concepts/details
   (Download pdf file from the above link)

Reference Books:

1. Erik Hellman, **"Android Programming – Pushing the Limits"**, 1st Edition. Wiley India Pvt Ltd. 2014. ISBN-13: 978-8126547197
2. Dawn Griffiths and David Griffiths, **"Head First Android Development"**, 1st Edition. O'Reilly SPD Publishers, 2015. ISBN-13:978-9352131341
3. Bill Phillips, Chris Stewart and Kristin Marsicano, **"Android Programming: The Big Nerd Ranch Guide"**, 3rd Edition, Big Nerd Ranch Guides, 2017. ISBN-13:978-0134706054

Alva's Institute of Engineering & Technology

Shobhavana Campus, Mijar, Moodbidri, D.K – 574225

Phone: 08258-262725, Fax: 08258-262726

DEPARTMENT OF AIRTIFICIAL INTELLIGENCE & MACHINE LEARNING

SEMESTER V

Course Code: **18AI63** | Course Name: **Java for Mobile Applications(JMA)**

Course Teacher: **Mr. Shrikanth N G**

Course Outcomes: After studying this course, students will be able to:

CO Numbers	Course Outcomes	Blooms Level	Target Level
18AI63.1	**Interpret** the need for advanced Java concepts like enumerations, Auto Boxing and annotations.	Understand (L2)	2
18AI63.2	**Demonstrate** the concept of Collections, Comparators, Legacy classes and Interfaces.	Understand (L2)	2
18AI63.3	**Illustrate** the use of string handling functions.	Understand (L2)	2
18AI63.4	**Demonstrate** the Android Platform, its architecture and features.	Understand (L2)	2
18AI63.5	**Design and Develop an** user interface, database application and content providers using Android.	Apply (L3)	2

CO-PO/CO-PSO Mapping Matrix:

CO Numbers	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4	PSO5
18AI63.1	2	2	2		2							2	2			2	2
18AI63.2	2	2	2		2							2	2			2	2
18AI63.3	2	2	2		2							2	2			2	2
18AI63.4	2	2	2		2							2	2			2	2
18AI63.5	2	2	2		2							2	2			2	2
Avg	**2**	**2**	**2**		**2**							**2**	**2**			**2**	**2**

CO-PO/CO-PSO Mapping Matrix Justification: Student should have

CO Numbers	Pos	Level	Justification
18AI63.1	PO1	2	Moderately mapped as students gain the knowledge advanced java concepts in writing the programs
	PO2	2	Moderately mapped as only few students identify their own problem by conducting literature review for writing programs
	PO3	2	Moderately mapped as designing and implementation is required to write the program for the given problem statement.

DEPARTMENT OF AIRTIFICIAL INTELLIGENCE & MACHINE LEARNING

	PO5	2	Moderately mapped as students learn modern tools and advanced concepts of java to execute java programs / applications.
	PO12	2	Moderately mapped as students apply the concepts learnt in continuing professional development and new developments.
	PSO1	2	Moderately mapped as students understand and analyze advanced concepts of java and fluent in the use of concepts in writing the programs to meet real world problems.
	PSO4	2	Moderately mapped as students need the knowledge of modern tools to provide the solutions to complex problems.
	PSO5	2	Moderately mapped as students need to develop the mini project and problem solving skills in java contributes to successful career development.
18AI63.2	PO1	2	Moderately mapped as the students need the knowledge of python syntax and semantics related Collections and comparators to apply them in building applications which needs java programming constructs
	PO2	2	Moderately mapped as student need analyze the problem for solving /developing any application using appropriate java programming construct such as Collections, Comparators, Legacy classes and Interfaces..
	PO3	2	Moderately mapped as the process of design and implementation has to be followed while applying the concepts.
	PO5	2	Moderately mapped as students learn modern tools and concepts of java like Collections, Comparators, Legacy classes and Interfaces to execute java programs / applications.
	PO12	2	Moderately mapped as students apply the concepts learnt in continuing professional development and new developments.
	PSO1	2	Moderately mapped as students need the knowledge of java related to Collections, Comparators, Legacy classes and Interfaces to apply them in writing the programs to meet real world problems.
	PSO4	2	Moderately mapped as students need the knowledge of modern tools to provide the solutions to complex problems.
	PSO5	2	Moderately mapped as students need to develop the mini project and problem solving skills in java contributes to successful career development.
18AI63.3	PO1	2	Moderately mapped as the students need the knowledge of string handling functions of java to build applications.
	PO2	2	Problem analysis is necessary for solving /developing any application using string handling functions
	PO3	2	Moderately mapped as the process of design and implementation has to be followed while applying the concepts.
	PO5	2	Moderately mapped as students learn modern tools to handle string handling functions to execute java programs / applications
	PO12	2	Moderately mapped as students apply the concepts learnt in continuing professional development and new developments.
	PSO1	2	Moderately mapped as students need the knowledge of string handling functions to meet real world problems.
	PSO4	2	Moderately mapped as students need the knowledge of modern tools to provide the solutions to complex problems.
	PSO5	2	Moderately mapped as students need to develop the mini project and problem solving skills in java contributes to successful career development.

DEPARTMENT OF AIRTIFICIAL INTELLIGENCE & MACHINE LEARNING

18AI63.4	PO1	2	Moderately mapped as the students need the knowledge of android architecture in java syntax and semantics and apply them in developing applications
	PO2	2	Moderately mapped to problem analysis as the moderate number of students can use the concepts of android in finding the solutions to the problem.
	PO3	2	Moderately mapped to design and development as the moderate number of students use java in building the applications
	PO5	2	Moderately mapped as students learn modern tools to execute java programs / applications.
	PO12	2	Moderately mapped as students apply the concepts learnt in continuing professional development and new developments.
	PSO1	2	Moderately mapped as students need the knowledge of android in writing the programs to meet real world problems.
	PSO4	2	Moderately mapped as students need the knowledge of modern tools to provide the solutions to complex problems.
	PSO5	2	Moderately mapped as students need to develop the mini project and problem solving skills in java using android contributes to successful career development.
18AI63.5	PO1	2	Moderately mapped as the students should have the complete basic knowledge of the user interface, database application and content providers using Android
	PO2	2	Moderately mapped as the students need to perform the complete problem analysis while applying in user interface, database application and content providers using Android .
	PO3	2	Moderately mapped as all students design and develop the applications using user interface, database application and content providers using Android .
	PO5	2	Moderately mapped as students learn modern user interface tools to build and execute java applications
	PO12	2	Moderately mapped as students apply the concepts learnt in continuing professional development and new product developments.
	PSO1	2	Moderately mapped as students should have the complete basic knowledge of the user interface, database application and content providers using Android to meet real world problems.
	PSO4	2	Moderately mapped as students need the knowledge of modern tools to provide the solutions to complex problems.
	PSO5	2	Moderately mapped as students need to develop the mini project and problem solving skills in java contributes to successful career development.

15/3/23
Course Teacher
Signature with date

15/3/23
IQAC Member
Signature with date

15/03/2023
IQAC Chairman
Signature with date

Alva's Institute of Engineering & Technology

Shobhavana Campus, Mijar, Moodbidri, D.K – 574225
Phone: 08258-262725, Fax: 08258-262726

DEPARTMENT OF AIRTIFICIAL INTELLIGENCE & MACHINE LEARNING

SEMESTER V

ourse Code:**18AIMP68**	Course Name: **Mobile Application Development Laboratory**

ourse Teacher: **Mr. Shrikanth N G**

ourse Outcomes: After studying this course, students will be able to:

CO Numbers	Course Outcomes	Blooms Level	Target Level
18AIMP68.1	Develop Android application by setting up Android development environment.	Apply (L3)	2
18AIMP68.2	Develop adaptive, responsive user interfaces that work across a wide range of devices	Apply (L3)	2
18AIMP68.3	Infer long running tasks and background work in Android applications.	Apply (L3)	2
18AIMP68.4	Demonstrate methods in storing, sharing and retrieving data in Android applications.	Apply (L3)	2
18AIMP68.5	Infer the role of permissions and security for Android applications.	Apply (L3)	2

O-PO/CO-PSO Mapping Matrix:

CO Numbers	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4	PSO5
8AIMP68.1	1	2	2	1	2	-	-	-	1	-	2	2	2	2		2	2
8AIMP68.2	2	2	2	1	2	-	-	-	-	-	2	2	2	2		2	2
8AIMP68.3	1	2	2	2	2	-	-	-	-	-	1	2	2	2		2	2
8AIMP68.4	2	2	2	2	2	-	-	-	-	-	2	2	2	2		2	2
8AIMP68.5	2	2	2	1	2	-	-	-	-	-	2	2	2	2		2	2
Avg	1.6	2	2	1.4	2				1		1.8	2	2	2		2	2

O-PO/CO-PSO Mapping Matrix Justification: Student should have

CO Numbers	Pos	Level	Justification
18AIMP68.1	PO1	1	Installing android development environment will helps to gain the engineering knowledge in low level

	PO2	2	Analysing the problem for solution development using android requires principles of mathematics in moderate level.
	PO3	2	In android creating the application will contribute in Design and development of solution in moderate level.
	PO4	1	Some android application development requires analysing the complex problems in low level.
	PO5	2	Creating android app requires the use of modern tools in moderate level.
	PO9	1	Designing the solution for the project will contribute in Individual and Team Work in Low Level.
	PO11	2	Creating the various solutions for the real time problems requires the managing the project work with finance in moderate level
	PO12	2	Mobile application development will contribute in lifelong learning in moder level.
	PSO1	2	Ability to understand and implement android applications for real time problems requires professional skills in moderate level
	PSO2	2	Analysing and implementing the AI and ML techniques for industry using android applications at moderate level.
	PSO4	2	Analysing and implementing the android app requires problem solving skills in moderate level.
	PSO5	2	Knowledge of developing android application will contribute successful career and Entrepreneurship in moderate level
18AIMP68.2	PO1	2	Creating the user interfaces for various problems requires engineering knowledge in moderate level.
	PO2	2	Implementing the responsive, adaptive user interface will contribute in analysing the problems in moderate level
	PO3	2	Developing the user interface for various societal applications will contribute in design and development of solution in moderate level.
	PO4	1	Interpretation of various health and safety related user interface requires the knowledge of investigation of complex problems in low level.
	PO5	2	Creating and applying the appropriate techniques for designing the variou user interfaces will contribute in modern tool usage in moderate level.
	PO11	2	Creating the various solutions for the real time problems requires the managing the project work with finance in moderate level
	PO12	2	Mobile application development will contribute in lifelong learning in moderate level.
	PSO1	2	Ability to understand and implement android applications for real time problems requires professional skills in moderate level
	PSO2	2	Analysing and implementing the AI and ML techniques for industry using android applications at moderate level.
	PSO4	2	Analysing and implementing the android app requires problem solving skills in moderate level.
	PSO5	2	Knowledge of developing android application will contribute successful career and Entrepreneurship in moderate level
18AIMP68.3	PO1	1	Constructing a long running or background working android app requires engineering knowledge in low level.

Alva's Institute of Engineering & Technology

Shobhavana Campus, Mijar, Moodbidri, D.K – 574225
Phone: 08258-262725, Fax: 08258-262726

DEPARTMENT OF AIRTIFICIAL INTELLIGENCE & MACHINE LEARNING

SEMESTER V

Course Code: **18AIMP68**

Course Name: **Mobile Application Development Laboratory**

Course Teacher: **Mr. Shrikanth N G**

Course Outcomes: After studying this course, students will be able to:

CO Numbers	Course Outcomes	Blooms Level	Target Level
18AIMP68.1	Develop Android application by setting up Android development environment.	Apply (L3)	2
18AIMP68.2	Develop adaptive, responsive user interfaces that work across a wide range of devices	Apply (L3)	2
18AIMP68.3	Infer long running tasks and background work in Android applications.	Apply (L3)	2
18AIMP68.4	Demonstrate methods in storing, sharing and retrieving data in Android applications.	Apply (L3)	2
18AIMP68.5	Infer the role of permissions and security for Android applications.	Apply (L3)	2

CO-PO/CO-PSO Mapping Matrix:

CO Numbers	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4	PSO5
18AIMP68.1	1	2	2	1	2	-	-	-	1	-	2	2	2	2		2	2
18AIMP68.2	2	2	2	1	2	-	-	-	-	-	2	2	2	2		2	2
18AIMP68.3	1	2	2	2	2	-	-	-	-	-	1	2	2	2		2	2
18AIMP68.4	2	2	2	2	2	-	-	-	-	-	2	2	2	2		2	2
18AIMP68.5	2	2	2	1	2	-	-	-	-	-	2	2	2	2		2	2
Avg	1.6	2	2	1.4	2				1		1.8	2	2	2		2	2

CO-PO/CO-PSO Mapping Matrix Justification: Student should have

CO Numbers	Pos	Level	Justification
18AIMP68.1	PO1	1	Installing android development environment will helps to gain the engineering knowledge in low level

	PO2	2	Analysing the problem for solution development using android requires principles of mathematics in moderate level.
	PO3	2	In android creating the application will contribute in Design and development of solution in moderate level.
	PO4	1	Some android application development requires analysing the complex problems in low level.
	PO5	2	Creating android app requires the use of modern tools in moderate level.
	PO9	1	Designing the solution for the project will contribute in Individual and Team Work in Low Level.
	PO11	2	Creating the various solutions for the real time problems requires the managing the project work with finance in moderate level
	PO12	2	Mobile application development will contribute in lifelong learning in moderate level.
	PSO1	2	Ability to understand and implement android applications for real time problems requires professional skills in moderate level
	PSO2	2	Analysing and implementing the AI and ML techniques for industry using android applications at moderate level.
	PSO4	2	Analysing and implementing the android app requires problem solving skills in moderate level.
	PSO5	2	Knowledge of developing android application will contribute successful career and Entrepreneurship in moderate level
18AIMP68.2	PO1	2	Creating the user interfaces for various problems requires engineering knowledge in moderate level.
	PO2	2	Implementing the responsive, adaptive user interface will contribute in analysing the problems in moderate level
	PO3	2	Developing the user interface for various societal applications will contribute in design and development of solution in moderate level.
	PO4	1	Interpretation of various health and safety related user interface requires the knowledge of investigation of complex problems in low level.
	PO5	2	Creating and applying the appropriate techniques for designing the various user interfaces will contribute in modern tool usage in moderate level.
	PO11	2	Creating the various solutions for the real time problems requires the managing the project work with finance in moderate level
	PO12	2	Mobile application development will contribute in lifelong learning in moderate level.
	PSO1	2	Ability to understand and implement android applications for real time problems requires professional skills in moderate level
	PSO2	2	Analysing and implementing the AI and ML techniques for industry using android applications at moderate level.
	PSO4	2	Analysing and implementing the android app requires problem solving skills in moderate level.
	PSO5	2	Knowledge of developing android application will contribute successful career and Entrepreneurship in moderate level
18AIMP68.3	PO1	1	Constructing a long running or background working android app requires engineering knowledge in low level.

Alva's Institute of Engineering & Technology

Shobhavana Campus, Mijar, Moodbidri, D.K – 574225

Phone: 08258-262725, Fax: 08258-262726

DEPARTMENT OF AIRTIFICIAL INTELLIGENCE & MACHINE LEARNING

SEMESTER V

Course Code: **18AIMP68**	Course Name: **Mobile Application Development Laboratory**

Course Teacher: **Mr. Shrikanth N G, Mrs. Rashmi Suvarna**

Course Outcomes: After studying this course, students will be able to:

CO Numbers	Course Outcomes	Blooms Level	Target Level
18AIMP68.1	Develop Android application by setting up Android development environment.	Apply (L3)	2
18AIMP68.2	Develop adaptive, responsive user interfaces that work across a wide range of devices	Apply (L3)	2
18AIMP68.3	Infer long running tasks and background work in Android applications.	Apply (L3)	2
18AIMP68.4	Demonstrate methods in storing, sharing and retrieving data in Android applications.	Apply (L3)	2
18AIMP68.5	Infer the role of permissions and security for Android applications.	Apply (L3)	2

CO-PO/CO-PSO Mapping Matrix:

CO Numbers	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4	PSO5
18AIMP68.1	1	2	2	1	2	-	-	-	1	-	2	2	2	2		2	2
18AIMP68.2	2	2	2	1	2	-	-	-	-	-	2	2	2	2		2	2
18AIMP68.3	1	2	2	2	2	-	-	-	-	-	1	2	2	2		2	2
18AIMP68.4	2	2	2	2	2	-	-	-	-	-	2	2	2	2		2	2
18AIMP68.5	2	2	2	1	2	-	-	-	-	-	2	2	2	2		2	2
Avg	1.6	2	2	1.4	2				1		1.8	2	2	2		2	2

CO-PO/CO-PSO Mapping Matrix Justification: Student should have

CO Numbers	Pos	Level	Justification
18AIMP68.1	PO1	1	Installing android development environment will helps to gain the engineering knowledge in low level

	PO2	2	Realizing background running applications will contribute in analysing the problems in moderate level
	PO3	2	Developing background running mobile application for various real world problems will contribute in design and development of solution in moderate level.
	PO4	2	Some long running task application development requires analysing the complex problems in moderate level.
	PO5	2	Building and relating the appropriate techniques for designing the various background tasks will contribute in modern tool usage in moderate level.
	PO11	1	Generating the various solutions for the real time background applications requires the managing the project work with finance in low level
	PO12	2	Mobile application for long running apps development will contribute in lifelong learning in moderate level.
	PSO1	2	Ability to understand and implement android applications for real time problems requires professional skills in moderate level
	PSO2	2	Analysing and implementing the AI and ML techniques for industry using android applications at moderate level.
	PSO4	2	Analysing and implementing the android app requires problem solving skills in moderate level.
	PSO5	2	Knowledge of developing android application will contribute successful career and Entrepreneurship in moderate level
18AIMP68.4	PO1	2	Building the android apps for storing and retrieving the data will helps to gain the engineering knowledge in moderate level
	PO2	2	Comprehending applications for loading and fetching the data from different database will contribute in analysing the problems in moderate level
	PO3	2	Developing a solution for real time high end problems contributes in gaining the knowledge of designing the solutions in moderate level.
	PO4	2	Understanding of various database related applications requires the knowledge of investigation of complex problems in moderate level.
	PO5	2	Creating and applying the appropriate techniques for designing the database related apps will contribute in modern tool usage in moderate level.
	PO11	2	Producing the various solutions for the real time Storing and retrieving problems requires the managing the project work with finance in moderate level
	PO12	2	Database base mobile application development will contribute in lifelong learning in moderate level.
	PSO1	2	Ability to understand and implement android applications for real time problems requires professional skills in moderate level
	PSO2	2	Analysing and implementing the AI and ML techniques for industry using android applications at moderate level.
	PSO4	2	Analysing and implementing the android app requires problem solving skills in moderate level.
	PSO5	2	Knowledge of developing android application will contribute successful career and Entrepreneurship in moderate level
18AIMP68.5	PO1	2	Creating an application to provide security for various societal applications requires engineering knowledge in moderate level.
	PO2	2	Grasping the knowledge of security related applications will contribute in analysing the problems in moderate level

ALVA'S
Education Foundation

Alva's Institute of Engineering & Technology

Shobhavana Campus, Mijar, Moodbidri, D.K – 574225
Phone: 08258-262725, Fax: 08258-262726

DEPARTMENT OF AIRTIFICIAL INTELLIGENCE & MACHINE LEARNING

PO3	2	Developing applications that provides security for various real world problems will contribute in design and development of solution in moderate level.
PO4	1	Some security related application development requires analysing the complex problems in low level.
PO5	2	Constructing and involving the appropriate techniques for designing the safety related will contribute in modern tool usage in moderate level.
PO11	2	Engendering the various solutions for the real time applications requires the managing the project work with finance in low level
PO12	2	Banking and Societal related applications developments will contribute in lifelong learning in moderate level.
PSO1	2	Ability to understand and implement android applications for real time problems requires professional skills in moderate level
PSO2	2	Analysing and implementing the AI and ML techniques for industr[illegible] android applications at moderate level.
PSO4	2	Analysing and implementing the android app requires problem solving skills in moderate level.
PSO5	2	Knowledge of developing android application will contribute successful career and Entrepreneurship in moderate level

15/3/23
Course Teacher
Signature with date

15/3/23
IQAC Member
Signature with date

15/03/2023
IQAC Chairman
Signature with date

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY
(Unit of Alva's Education Foundation (R), Moodbidri)
Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi. Recognized by Government of Karnataka.
A+, Accredited by NAAC
Shobhavana Campus, MIJAR-574225, Moodbidri, D.K., Karnataka
Ph: 08258-262725; Mob:722262724,7026262725,mail:principalaiet08@gmail.com

Artificial Intelligence and Machine Learning

JAVA FOR MOBILE APPLICATIONS
SEMESTER – VI
AY 2022-23 EVEN

Course Code-18AI63

Faculty Name : Mr. Shrikanth N G

Content beyond the syllabus

- Exception handling fundamentals, types and uncaught exceptions.
- Multithreading programming.

Course Coordinator

Signature of IQAC Member(Module)

Signature of IQAC Chairman(HOD)

s? With an example code, demonstrat

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Unit of Alva's Education Foundation (R), Moodbidri)
Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi. Recognized by Government of Karnataka.
A+, Accredited by NAAC
Shobhavana Campus, MIJAR-574225, Moodbidri, D.K., Karnataka
Ph: 08258-262725; Mob:722262724,7026262725,mail:principalaiet08@gmail.com

Artificial Intelligence and Machine Learning

JAVA FOR MOBILE APPLICATIONS
SEMESTER – VI
AY 2022-23 EVEN

Course Code-18AI63

Faculty Name : Mr. Shrikanth N G

Modes of content delivery

Module No	Modes of content delivery
I	• Lecture through black board • Lecture through presentation
II	• Lecture through black board • Lecture through presentation
III	• Lecture through black board • Lecture through presentation • Demonstration Class
IV	• Lecture through black board • Lecture through presentation • Flip Class
V	• Lecture through black board • Lecture through presentation • Demonstration Class

Course Coordinator

Signature of IQAC Member(Module)

Signature of IQAC Chairman(HOD)

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Unit of Alva's Education Foundation (R), Moodbidri)
Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi. Recognized by Government of Karnataka.
A+, Accredited by NAAC
Shobhavana Campus, MIJAR-574225, Moodbidri, D.K., Karnataka
Ph: 08258-262725; Mob:722262724,7026262725,mail:principalaiet08@gmail.com

ALVA'S
Education Foundation®

Artificial Intelligence and Machine Learning

JAVA FOR MOBILE APPLICATIONS
SEMESTER – VI
AY 2022-23 EVEN

Course Code-18AI63

Faculty Name : Mr. Shrikanth N G

Modes of delivery for content beyond the syllabus

Topic or Module Name	Modes of content delivery
Exception handling fundamentals, types and uncaught exceptions.	• Lecture through presentation • Lecture through demonstration
Multithreading programming.	• Lecture through presentation • Lecture through demonstration

Course Coordinator

Signature of IQAC Member(Module)

Signature of IQAC Chairman(HOD)

USN

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Accredited by NAAC with A+ Grade)

Department of Artificial Intelligence and Machine Learning

Continuous Internal Evaluation Test I ,AY 2022-23

Course Title : **Java for Mobile Applications**		Course Code: 18AI63
Date: 26/5/2023	Time: 9.30 AM- 11.00 AM	Semester/Section: VI
Faculty: Mr. Shrikanth N G		Max. Marks: 30

Note: Answer ONE FULL question from each Module.

Q. No.		Questions	Marks	COs	BTL
		Module 3			
1	a)	What are enumerations? With an example code, demonstrate how varieties of mangoes can be represented through enumeration.	8	CO1	L2
	b)	Give the explanation why Wrapper classes are required when compared to primitive types.	7	CO1	L2
		OR			
2	a)	Demonstrate with a Java code, how Auto boxing/Unboxing Occurs in Expressions?	8	CO1	L2
	b)	How default values can be used in an Annotations? Explain with an example Java Code.	7	CO1	L2
		Module 5			
3	a)	Briefly discuss the various Collection framework interfaces with the methods declared in it.	10	CO2	L2
	b)	Write a Java program to demonstrate the use of values () and value Of () methods.	5	CO1	L2
		OR			
4	a)	List and explain the advantages of Collections Framework in developing a generic Java program.	10	CO2	L2
	b)	How Auto boxing/Unboxing can be used to prevent errors? Explain.	5	CO1	L2

Levels of Bloom's Taxonomy

No.	**L1**	**L2**	**L3**	**L4**	**L5**	**L6**
Level	Remember	Understand	Apply	Analyze	Evaluate	Create

Course Outcomes

CO1	**Interpret** the need for advanced Java concepts like enumerations, Auto Boxing and annotations.
CO2	**Demonstrate** the concept of Collections, Comparators, Legacy classes and Interfaces.
CO3	**Illustrate** the use of string handling functions.
CO4	**Demonstrate** the Android Platform, its architecture and features.
CO5	**Design and Develop** an user interface, database application and content providers using Android.

ALVA'S
Education Foundation®

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Accredited by NAAC with A+ Grade)

QUESTION PAPER REVIEW REPORT

Continuous Internal Evaluation (CIE) Test:IIIA Y 2022-23

Department : AIML

Semester/Section:6th / A

Max Marks: 30

Course Title: Java For Mobile Applications

Course Code:18AI63

Date:26/5/23

Faculty: Shrikanth N G

Qn. No.	Course Outcome (CO)	Bloom's Taxonomy Level	Marks
1a	CO1	L2	8
1b	CO1	L2	7
2a	CO1	L2	8
2b	CO1	L2	7
3a	CO2	L2	10
3b	CO1	L2	5
4a	CO2	L2	10
4b	CO1	L2	5
Total Marks			60

BT Level: L1-Remember, L2-Understand, L3 -Apply, L4 -Analyze, L5- Evaluate, L6- Create

Consolidated Marks for Different BT Levels:

BT Level	Marks for Each Level	% of Marks	Remarks
L2	60	100	

Scrutinizer/Reviewer Remark:

Approved		Approved with Correction		Rejected	
Reason for Rejection					

Kiran Raj K M

Name & Signature of the Scrutinizer

Date: 23|5|2023

SHRIKANTH N.G.

Name & Signature of the IQAC Coordinator

Date: 23|5|23

Signature of Head of the Department

20|01|2023

USN □□□□□□□□□□

ALVA'S INSTITUTE OF ENGINEERING&TECHNOLOGY, MOODBIDRI

(A unit of Alva's Education Foundation)

Sixth Semester B.E. (AIML) I. A Test Examinations-I

26th May – 2023

18AI63 – JAVA FOR MOBILE APPLICATIONS

SCHEME OF VALUATION

Questions		Details	Marks

1	a.	Enumerators contain a list of constant values that apply to a certain type of data, or object.
		• They can be useful in setting a scope of values for a particular object.
		• An enumeration defines a class type.
		• An enumeration can have constructors, methods, and instance variables.
		• An enum is actually a new type of class.
		• You can declare them as inner classes or outer classes.
		• You can declare variables of an enum type.
		• Each declared value is an instance of the enum class.
		• Enums are implicitly public, static, and final.
		
		enum Mangoes {
		raspuri,badam;
		}
		class Enu
		{
		public static void main(String args[])
		
		{
		Mangoes S;
		S=Mangoes.*raspuri*;
		System.*out*.println("Gender IS="+S);
		}
		}
		
		Explanation of above code
			2.5M
			3M
			2.5M

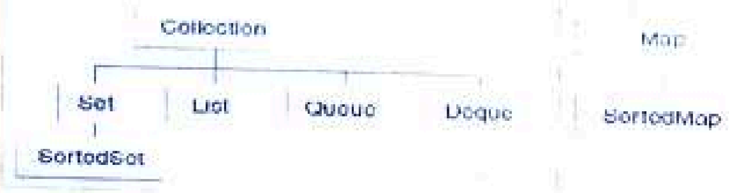
1	b.	Java uses primitive types (also called simple types), such as int or double, to hold the basic data types supported by the language. • Instead of primitive types if objects are used everywhere for even simple calculations then performance overhead is the problem. • So to avoid this java had used primitive types. • So primitive types do not inherit Object class	2M

		• But there are times when you will need an object representation for primitives like int and char. • Example, you can't pass a primitive type by reference to a method. • Many of the standard data structures implemented by Java operate on objects, which mean that you can't use these data structures to store primitive types. • To handle these (and other) situations, Java provides *type wrappers, which are classes* that encapsulate a primitive type within an object. Example of Wrapper Classes with explanation or Object functionality Nullability More generic Advanced functionality	3M+2M or 2M

2	a	autoboxing and unboxing take place whenever a conversion into an object or from an object is required. This applies to expressions. Within an expression, a numeric object is automatically unboxed. The outcome of the expression is reboxed, if necessary
		
		class auto {
		public static void main(String args[]) { Integer iOb, iOb2; **int** i;
		iOb = 100;
		System.*out*.println("Original value of iOb: " + iOb); The following automatically unboxes iOb, performs the increment, and [illegible]
		++iOb;
		System.*out*.println("After ++iOb: " + iOb); iOb2 = iOb + (iOb / 3);
		System.*out*.println("iOb2 after expression: " + iOb2); i = iOb + (iOb / 3);
		System.*out*.println("i after expression: " + i);
		}
		}
		output: Original value of iOb: 100 After ++iOb: 101
		iOb2 after expression: 134 i after expression: 134
		
		Explanation of above code
			2M
			
			4M
			
			2M

2	b	we can give annotation members default values that will be used if no value is specified when the annotation is applied. A default value is specified by adding a **default** clause to a member's declaration. It has this general form: *type member*( ) default *value* ; **import** java.lang.annotation.*; **import** java.lang.reflect.*; **import** java.lang.annotation.*; @Retention(RetentionPolicy.*RUNTIME*) **@interface** MyINF { String str() **default** "WELCOME";	1M

		int val() default 5;
		}
		class annu { @MyINF()
		public static void myMeth() { annu ob = **new** annu();
		try {
		Class<?> c = ob.getClass();
		Method m = c.getMethod("myMeth");
		MyINF anno = m.getAnnotation(MyINF.**class**); System.*out*.println(anno.str() + " " + anno.val());
		}
		catch (NoSuchMethodException exc) { System.*out*.println("Method Not Found"); }
		}
		}
		public static void main(String args[]) {
		myMeth();
		}
		}
		
		Explanation of above code
			5M
			
			1M

3	a	 Collection Set List Queue Deque SortedSet Map SortedMap Methods of above Collection Interfaces	5*2=10 M

3	b	Method returns an array that contains a list of the enumeration constants .
		values() returns the values in the enumeration and stores them in an array. We can process the array with a foreach loop.
		Systex: public static *enum-type[ ] values( )*
		enum Days {
		mon,tue,wed,thu,fri,sat,sun;
		}
		class cont
		{
		public static void main(String args[])
		{
		Days d[]=Days.*values*();
		for(Days d1:d)
		System.*out*.println("today day is:"+d1);
		}
		}
		
		method takes a single parameter of the constant name to retrieve and returns the constant from the enumeration, if it exists.
			2.5M

		method returns the enumeration constant whose value corresponds to the string passed in *str*.
		Syntax: enumerationVariable = enumerationName.valueOf("EnumerationValueInList");
		enum Days {
		WeekDays wd = WeekDays.valueOf("MONDAY"); System.out.println(wd);
		monday,tuesday;
		}
		class cont
		{
		public static void main(String args[])
		{
		Days d=Days.*valueOf*("monday"); System.*out*.println("day selected is:"+d);
		}
		}
			2.5M

4	a	Reduces programming effort: **Increases program speed and quality :** **Allows interoperability among unrelated APIs:** Reduces effort to learn and to use new APIs Reduces effort to design new APIs: Fosters software reuse: Explanation of above points	1M 9M

4	b	• Autoboxing always creates the proper object and auto unboxing always produce the proper value.
		• There is no wayfor the process to produce the wrong type of object or value. Program:
		class auto {
		public static void main(String args[]) {
		Integer iOb = 1000; // autobox the value 1000
		int i = iOb.byteValue(); // manually unbox as byte !!! System.*out*.println("unbox value:"+i); // display 1000 !
		}
		}
		output:
		unbox value:-24
		• This program displays not the expected value of 1000, but –24! The reason is that the value inside iOb is manually unboxed by calling byteValue( ), which causes the truncation of the value stored in iOb, which is 1,000.
		• This results in the garbage value of –24 being assigned to i.
		• Auto-unboxing prevents this type of error because the value in iOb will always autounbox into a value compatible with int.
			5M

Signature of Faculty

Signature of HOD

USN □□□□□□□□□□

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Accredited by NAAC with A+ Grade)

Department of Artificial Intelligence and Machine Learning

Continuous Internal Evaluation Test II ,AY 2022-23

Course Title : **Java for Mobile Applications**		Course Code: 18AI63
Date: 19/6/2023	Time: 9.30 AM- 11.00 AM	Semester/Section: VI
Faculty: Mr. Shrikanth N G		Max. Marks: 30

Note: Answer ONE FULL question from each Module.

Q. No.		Questions	Marks	COs	BTL
		Module 3			
1	a)	With example code(include minimum 5 with explanation), discuss the various algorithm(Minimum 5) supported in Collections.	10	CO 2	L2
	b)	How Strings can compared with following method? Explain with example. i) equals and equalsIgnoreCase() ii) regionMatches	5	CO 3	L2
		OR			
2	a)	Explain the HashMap Class and TreeMap class in details with example program	10	CO 2	L2
	b)	How following methods can be used in character extraction? Explain with example. i) charAT() ii) getchars() iii)getBytes()	5	CO 3	L2
		Module 5			
3	a)	Explain different constructors supported by String class along with example code which includes all those constuctors.	10	CO 3	L2
	b)	Demonstrate the following string operations. Explain with example i) String Literals ii) String Concatenation iii) String Concatenation with other data types	5	CO 3	L2
		OR			
	a)	With relevant Program/ code example, explain the following StringBuffer methods i) length()and capacity() ii) setLength() iii) charAt() and setCahrAt() iv) reverse() v) insert()	10	CO 3	L2
	b)	Demonstrate the following string operations i)startsWith and endsWith() ii) String Conversion and toString() iii) indexOf( ) and lastIndexOf().	5	CO 3	L2

Levels of Bloom's Taxonomy

No.	L1	L2	L3	L4	L5	L6
Level	Remember	Understand	Apply	Analyze	Evaluate	Create

Course Outcomes

CO1	**Interpret** the need for advanced Java concepts like enumerations, Auto Boxing and annotations.
CO2	**Demonstrate** the concept of Collections, Comparators, Legacy classes and Interfaces.
CO3	**Illustrate** the use of string handling functions.
CO4	**Demonstrate** the Android Platform, its architecture and features.
CO5	**Design and Develop an** user interface, database application and content providers using Android.

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Accredited by NAAC with A+ Grade)

QUESTION PAPER REVIEW REPORT

Continuous Internal Evaluation (CIE) Test: IIAY 2022-23

Department : AIML Semester/Section:6th / A Max Marks: 30

Course Title: Java For Mobile Applications Course Code:18AI63 Date:19/6/23

Faculty: Shrikanth N G

Qn. No.	Course Outcome (CO)	Bloom's Taxonomy Level	Marks
1a	CO2	L2	10
1b	CO3	L2	5
2a	CO2	L2	10
2b	CO3	L2	5
3a	CO3	L2	10
3b	CO3	L2	5
4a	CO3	L2	10
4b	CO3	L2	5
Total Marks			60

BT Level: L1-Remember, L2-Understand, L3 -Apply, L4 -Analyze, L5- Evaluate, L6- Create

Consolidated Marks for Different BT Levels:

BT Level	Marks for Each Level	% of Marks	Remarks
L2	60	100	

Scrutinizer/Reviewer Remark:

Approved		Approved with Correction		Rejected	
Reason for Rejection					

Kiran Raj K.M

Name & Signature of the Scrutinizer

Date: 17/6/23

SHRIKANTH N.G.

Name & Signature of the IQAC Coordinator

Date: 17/6/23

17/06/2023

Signature of Head of the Department

ALVA'S INSTITUTE OF ENGINEERING&TECHNOLOGY, MOODBIDRI

(A unit of Alva's Education Foundation)

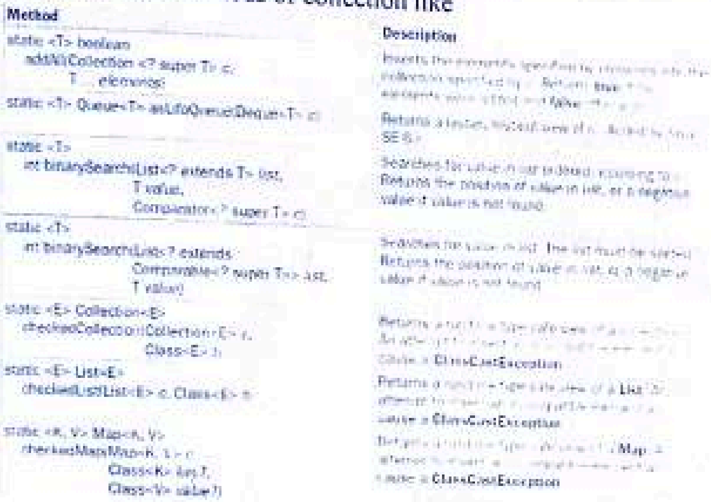
Sixth Semester B.E. (AIML) I. A Test Examinations-II

19th June – 2023

18AI63 – JAVA FOR MOBILE APPLICATIONS

SCHEME OF VALUATION

<table>
<tr><th colspan="2">Questions</th><th>Details</th><th>Marks</th></tr>
<tr><td rowspan="2"></td><td rowspan="2">a.</td><td>Any five Static methods of collection like



Method	Description
static <T> boolean addAll(Collection <? super T> c, T ... elements)	Inserts the elements specified by elements into the collection specified by c. Returns true if the elements were added and false otherwise.
static <T> Queue<T> asLifoQueue(Deque<T> c)	Returns a last-in, first-out view of c. (Added by Java SE 6.)
static <T> int binarySearch(List<? extends T> list, T value, Comparator<? super T> c)	Searches for value in list ordered according to c. Returns the position of value in list, or a negative value if value is not found.
static <T> int binarySearch(List<? extends Comparable<? super T>> list, T value)	Searches for value in list. The list must be sorted. Returns the position of value in list, or a negative value if value is not found.
static <E> Collection<E> checkedCollection(Collection<E> c, Class<E> t)	Returns a run-time type-safe view of a collection. An attempt to insert an incompatible element will cause a ClassCastException.
static <E> List<E> checkedList(List<E> c, Class<E> t)	Returns a run-time type-safe view of a List. An attempt to insert an incompatible element will cause a ClassCastException.
static <K, V> Map<K, V> checkedMap(Map<K, V> c, Class<K> keyT, Class<V> valueT)	Returns a run-time type-safe view of a Map. An attempt to insert an incompatible element will cause a ClassCastException.

</td><td>5M</td></tr>
<tr><td>Program including 5algorithms with explanation of how it works in program</td><td>5M</td></tr>
<tr><td>1</td><td>b.</td><td>i) equals and equalsIgnoreCase()

boolean equals(Object *str*)
Here, *str* is the **String** object being compared with the invoking **String** object. It returns **true** if the strings contain the same characters in the same order, and **false** otherwise. The comparison is case-sensitive.
To perform a comparison that ignores case differences, call **equalsIgnoreCase( )**. When it compares two strings, it considers A-Z to be the same as a-z. It has this general form:
boolean equalsIgnoreCase(String *str*)
Here, *str* is the **String** object being compared with the invoking **String** object.
Program demonstrating the same

ii) ii) regionMatches
The **regionMatches( )** method compares a specific region inside a string with another specific region in another string. There is an overloaded form that allows you to ignore case in such</td><td>2.5M</td></tr>
</table>

		comparisons. Here are the general forms for these two methods: boolean regionMatches(int *startIndex*, String *str2*, int *str2StartIndex*, int *numChars*) boolean regionMatches(boolean *ignoreCase*, int *startIndex*, String *str2*, int *str2StartIndex*, int *numChars*) **Program demonstrating the same**	2.5M
2	a	The **HashMap** class extends **AbstractMap** and implements the **Map** interface. It uses a hash table to store the map. This allows the execution time of **get( )** and **put( )** to remain constant even for large sets. **HashMap** is a generic class that has this declaration: class HashMap<K, V> Here, **K** specifies the type of keys, and **V** specifies the type of values. The following constructors are defined: HashMap( ) HashMap(Map<? extends K, ? extends V> *m*) HashMap(int *capacity*) HashMap(int *capacity*, float *fillRatio*) The first form constructs a default hash map. The second form initializes the hash map by using the elements of *m*. The third form initializes the capacity of the hash map to *capacity*. The fourth form initializes both the capacity and fill ratio of the hash map by using its arguments. The meaning of capacity and fill ratio is the same as for **HashSet**, described earlier. The default capacity is 16. The default fill ratio is 0.75. **HashMap** implements **Map** and extends **AbstractMap**. It does not add any methods of its own. Program which demonstrate the HashMap. The **TreeMap** class extends **AbstractMap** and implements the **NavigableMap** interface. It creates maps stored in a tree structure. A **TreeMap** provides an efficient means of storing key/value pairs in sorted order and allows rapid retrieval. You should note that, unlike a hash map, a tree map guarantees that its elements will be sorted in ascending key order. **TreeMap** is a generic class that has this declaration: class TreeMap<K, V> Here, **K** specifies the type of keys, and **V** specifies the type of values. The following **TreeMap** constructors are defined: TreeMap( ) TreeMap(Comparator<? super K> *comp*) TreeMap(Map<? extends K, ? extends V> *m*) TreeMap(SortedMap<K, ? extends V> *sm*) The first form constructs an empty tree map that will be sorted by using the natural order of its keys. The second form constructs an empty tree-based map that will be sorted by using the **Comparator** *comp*. (Comparators are discussed later in this chapter.) The third form initializes a tree map with the entries from *m*, which will be sorted by using the natural order of the keys. The fourth form initializes a tree map with the entries from *sm*, which will be sorted in the same order as *sm*. **TreeMap** has no methods beyond those specified by the **NavigableMap** interface and	2.5M 2.5M 2.5M

<table>
<tr><td></td><td></td><td>the <b>AbstractMap</b> class

Program which demonstrate the TreeMap.</td><td>2.5M</td></tr>
<tr><td>2</td><td>b</td><td>i) charAT()
To extract a single character from a <b>String</b>, you can refer directly to an individual character via the <b>charAt( )</b> method. It has this general form:
char charAt(int <i>where</i>)
Here, <i>where</i> is the index of the character that you want to obtain. The value of <i>where</i> must be nonnegative and specify a location within the string. <b>charAt( )</b> returns the character at the specified location. For example,<pre>
char ch;
ch = "abc".charAt(1);
</pre>assigns the value "b" to ch.</td><td>1.5M</td></tr>
<tr><td></td><td></td><td>ii) getchars()
If you need to extract more than one character at a time, you can use the <b>getChars( )</b> method. It has this general form:
void getChars(int <i>sourceStart</i>, int <i>sourceEnd</i>, char <i>target</i>[ ], int <i>targetStart</i>)
Here, <i>sourceStart</i> specifies the index of the beginning of the substring, and <i>sourceEnd</i> specifies an index that is one past the end of the desired substring.<pre>
class getCharsDemo {
public static void main(String args[]) {
String s = "This is a demo of the getChars method.";
int start = 10;
int end = 14;
char buf[] = new char[end - start];
s.getChars(start, end, buf, 0);
System.out.println(buf);
}
        }
</pre></td><td>2M</td></tr>
<tr><td></td><td></td><td>iii) getBytes()
There is an alternative to <b>getChars( )</b> that stores the characters in an array of bytes. This method is called <b>getBytes( )</b>, and it uses the default character-to-byte conversions provided by the platform. Here is its simplest form:
byte[ ] getBytes( )
Other forms of <b>getBytes( )</b> are also available. <b>getBytes( )</b> is most useful when you are exporting a <b>String</b> value into an environment that does not support 16-bit Unicode characters. For example, most Internet protocols and text file formats use 8-bit ASCII for all text interchange.</td><td>1.5M</td></tr>
<tr><td>3</td><td>a</td><td>To create an empty String, you call the default constructor.
<b>String(); String s = new String();</b>
To create strings that have initial values by an array of characters,
<b>String(char chars[ ])</b>
To specify a sub-range of a character array as an initializer using
<b>String(char chars[ ], int startIndex, int numChars)</b>
To construct a String object that contains the same character sequence as another String object using,
<b>String(String strObj)</b>
The String class provides constructors that initialize a string when given a byte array. Their forms are</td><td>5M</td></tr>
</table>

shown here

String(byte asciiChars[ ])

String(byte asciiChars[ ], int startIndex, int numChars

To construct a String from a StringBuffer by using the constructor.

String(StringBuffer *strBufObj*

Program demonstration of above constructors

5M

3 b

i) String Literals

1.5M

The earlier examples showed how to explicitly create a **String** instance from an array of characters by using the **new** operator. However, there is an easier way to do this using a string literal. For each string literal in your program, Java automatically constructs a **String** object. Thus, you can use a string literal to initialize a **String** object. For example, the following code fragment creates two equivalent strings:

```
char chars[] = { 'a', 'b', 'c' };
String s1 = new String(chars);
          String s2 = "abc"; // use string literal
```

ii) String Concatenation

1.5M

The one exception to this rule is the **+** operator, which concatenates two strings, producing a **String** object as the result. This allows you to chain together a series of **+** operations. For example, the following fragment concatenates three strings:

```
String age = "9";
String s = "He is " + age + " years old.";
System.out.println(s);
```

iii) String Concatenation with other data types

2M

You can concatenate strings with other types of data. For example, consider this slightly different version of the earlier example:

```
int age = 9;
String s = "He is " + age + " years old.";
System.out.println(s);
```

In this case, **age** is an **int** rather than another **String**, but the output produced is the same as before. This is because the **int** value in age is automatically converted into its string representation within a **String** object. This string is then concatenated as before. The compiler will convert an operand to its string equivalent whenever the other operand of the **+** is an instance of **String**.

Be careful when you mix other types of operations with string concatenation expressions, however. You might get surprising results. Consider the following:

```
String s = "four: " + 2 + 2;
          System.out.println(s);
```

4 a

i) length() and capacity()

The current length of a **StringBuffer** can be found via the **length( )** method, while the total allocated capacity can be found through the **capacity( )** method. They have the following general forms:

int length( )

int capacity( )

Here is an example:

```
// StringBuffer length vs. capacity.
class StringBufferDemo {
public static void main(String args[]) {
```

```
StringBuffer sb = new StringBuffer("Hello");
System.out.println("buffer = " + sb);
System.out.println("length = " + sb.length());
System.out.println("capacity = " + sb.capacity());
}
          }
```

ii) setLength()

To set the length of the buffer within a **StringBuffer** object, use **setLength( )**. Its general form is shown here:

void setLength(int *len*)

Here, *len* specifies the length of the buffer. This value must be nonnegative. When you increase the size of the buffer, null characters are added to the end of the existing buffer. If you call **setLength( )** with a value less than the current value returned by **length( )**, then the characters stored beyond the new length will be lost. The **setCharAtDemo** sample program in the following section uses **setLength( )** to shorten a **StringBuffer**.

5*2=10 M

iii) charAt() and setCharAt()

The value of a single character can be obtained from a **StringBuffer** via the **charAt( )** method. You can set the value of a character within a **StringBuffer** using **setCharAt( )**. Their general forms are shown here:

char charAt(int *where*)

void setCharAt(int *where*, char *ch*)

For **charAt( )**, *where* specifies the index of the character being obtained. For **setCharAt( )**, *where* specifies the index of the character being set, and *ch* specifies the new value of that character. For both methods, *where* must be nonnegative and must not specify a location beyond the end of the buffer.

The following example demonstrates **charAt( )** and **setCharAt( )**:

```
// Demonstrate charAt() and setCharAt().
class setCharAtDemo {
public static void main(String args[]) {
StringBuffer sb = new StringBuffer("Hello");
System.out.println("buffer before = " + sb);
System.out.println("charAt(1) before = " + sb.charAt(1));
sb.setCharAt(1, 'i');
sb.setLength(2);
System.out.println("buffer after = " + sb);
System.out.println("charAt(1) after = " + sb.charAt(1));
}
          }
```

iv) reverse()

You can reverse the characters within a **StringBuffer** object using **reverse( )**, shown here:

StringBuffer reverse( )

This method returns the reversed object on which it was called. The following program demonstrates **reverse( )**:

```
// Using reverse() to reverse a StringBuffer.
class ReverseDemo {
public static void main(String args[]) {
StringBuffer s = new StringBuffer("abcdef");
System.out.println(s);
s.reverse();
System.out.println(s);
}
        }
```

v) insert()

The **insert( )** method inserts one string into another. It is overloaded to accept values of all the simple types, plus **Strings**, **Objects**, and **CharSequences**. Like **append( )**, it calls **String.valueOf( )** to obtain the string representation of the value it is called with. This string is then inserted into the invoking **StringBuffer** object. These are a few of its forms:

StringBuffer insert(int *index*, String *str*)
StringBuffer insert(int *index*, char *ch*)
StringBuffer insert(int *index*, Object *obj*)

Here, *index* specifies the index at which point the string will be inserted into the invoking **StringBuffer** object.

The following sample program inserts "like" between "I" and "Java":

```
// Demonstrate insert().
class insertDemo {
public static void main(String args[]) {
StringBuffer sb = new StringBuffer("I Java!");
sb.insert(2, "like ");
System.out.println(sb);
}
        }
```

4 b i)startsWith and endsWith()

String defines two routines that are, more or less, specialized forms of **regionMatches( )**. The **startsWith( )** method determines whether a given **String** begins with a specified string. Conversely, **endsWith( )** determines whether the **String** in question ends with a specified string. They have the following general forms: 1.5M

boolean startsWith(String *str*)
boolean endsWith(String *str*)

Here, *str* is the **String** being tested. If the string matches, **true** is returned. Otherwise, **false** is returned. For example,

```
"Foobar".endsWith("bar")
```

and

```
"Foobar".startsWith("Foo")
```

ii)String Conversion and toString()

For the simple types, **valueOf( )** returns a string that contains the human-readable equivalent of the value with which it is called. For objects, **valueOf( )** calls the **toString( )** method on the object. We will look more closely at **valueOf( )** later in this chapter. Here, let's examine the **toString( )** method, because it is the means by which you can determine the string representation for objects of classes that you create.

Every class implements **toString( )** because it is defined by **Object**. However, the default implementation of **toString( )** is seldom sufficient. For most important classes that you create, you will want to override **toString( )** and provide your own string representations. Fortunately, this is easy to do. The **toString( )** method has this general form: 2M

String toString( )

To implement **toString( )**, simply return a **String** object that contains the human-readable string that appropriately describes an object of your class.

iii)indexOf() and lastIndexOf( )

The **String** class provides two methods that allow you to search a string for a specified character or substring:

- **indexOf( )** Searches for the first occurrence of a character or substring.
- **lastIndexOf( )** Searches for the last occurrence of a character or substring.

```
class indexOfDemo {
public static void main(String args[]) {
String s = "Now is the time for all good men " +
```

1.5M

		"to come to the aid of their country.";
		System.out.println(s);
		System.out.println("indexOf(t) = " +s.indexOf('t'));
		System.out.println("lastIndexOf(t) = " +s.lastIndexOf('t'));}}	

Signature of Faculty

Signature of HOD

USN □□□□□□□□□□

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Accredited by NAAC with A+ Grade)

Department of Artificial Intelligence and Machine Learning

Continuous Internal Evaluation Test III ,AY 2022-23

Course Title : **Java for Mobile Applications**		Course Code: 18AI63
Date: 6/7/2023	Time: 3.00 PM- 4.30 PM	Semester/Section: VI
Faculty: Mr. Shrikanth N G		Max. Marks: 30

Note: Answer ONE FULL question from each Module.

Q. No.		Questions	Marks	COs	BTL
		Part A			
1	a)	What is an Activity? With a neat diagram explain the Activity life Cycle.	8	CO4	L2
	b)	What are Intents? With a Java code, demonstrate how intents can be used to i) Switch between Activities ii) to start an activity for result	7	CO4	L2
		OR			
2	a)	Explain with a Java code to i) Pass Data Using an Intent Object ii) display a progress dialog	8	CO4	L2
	b)	Which are the states, a fragments goes through after it's creation. List the different methods that are called when fragment transits from one state to another.	7	CO4	L2
		Part B			
3	a)	What are the different layouts available to design user interface of an Android Applications? Justify the use of each layout.	8	CO5	L2
	b)	Write a Java Code to build a Quiz Application by using RadioGroup Class. Consider a suitable view for designing the front end.	7	CO5	L2
		OR			
4	a)	With the relevant code snippet, explain the use of following views i) Checkbox ii) ToggleButton iii) ImageButton iv) EditText	8	CO5	L2
	b)	Demonstrate how CRUD operations can be performed programmatically in Android application	7	CO5	L2

Levels of Bloom's Taxonomy

No.	L1	L2	L3	L4	L5	L6
Level	Remember	Understand	Apply	Analyze	Evaluate	Create

Course Outcomes

CO1	**Interpret** the need for advanced Java concepts like enumerations, Auto Boxing and annotations.
CO2	**Demonstrate** the concept of Collections, Comparators, Legacy classes and Interfaces.
CO3	**Illustrate** the use of string handling functions.
CO4	**Demonstrate** the Android Platform, its architecture and features.
CO5	**Design and Develop an** user interface, database application and content providers using Android.

QUESTION PAPER REVIEW REPORT

Continuous Internal Evaluation (CIE) Test:IIIAY 2022-23

Department : AIML

Semester/Section:6th / A

Max Marks: 30

Course Title: Java For Mobile Applications

Course Code:18AI63

Date:6/7/23

Faculty: Shrikanth N G

Qn. No.	Course Outcome (CO)	Bloom's Taxonomy Level	Marks
1a	CO4	L2	8
1b	CO4	L2	7
2a	CO4	L2	8
2b	CO4	L2	7
3a	CO5	L2	8
3b	CO5	L2	7
4a	CO5	L2	8
4b	CO5	L2	7
Total Marks			60

BT Level: L1-Remember, L2-Understand, L3 -Apply, L4 -Analyze, L5- Evaluate, L6- Create

Consolidated Marks for Different BT Levels:

BT Level	Marks for Each Level	% of Marks	Remarks
L2	60	100	

Scrutinizer/Reviewer Remark:

Approved		Approved with Correction		Rejected	
Reason for Rejection					

Kiran Raj K.M

Name & Signature of the Scrutinizer

Date: 3/7/23

SHRIKANTH N.G.

Name & Signature of the IQAC Coordinator

Date: 3/7/23

03/07/2023

Signature of Head of the Department

ALVA'S INSTITUTE OF ENGINEERING&TECHNOLOGY, MOODBIDRI

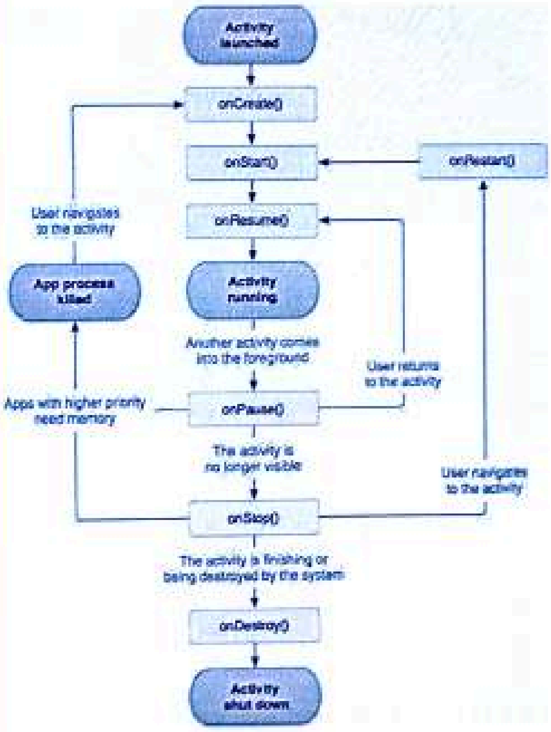
(A unit of Alva's Education Foundation)

Sixth Semester B.E. (AIML) I. A Test Examinations-III

6th July – 2023

18AI63 – JAVA FOR MOBILE APPLICATIONS

SCHEME OF VALUATION

Questions		Details	Marks
1	a.	An activity in Android is a specific combination of XML files and JAVA files. It is basically a container that contains the design as well as coding stuff. XML files provide the design of the screen and JAVA files deal with all coding stuff like handles, what is happening, design files, etc.  Activity launched onCreate() onStart() onRestart() User navigates to the activity onResume() App process killed Activity running Another activity comes into the foreground User returns to the activity Apps with higher priority need memory onPause() The activity is no longer visible User navigates to the activity onStop() The activity is finishing or being destroyed by the system onDestroy() Activity shut down **onCreate()** This is the first callback and called when the activity is first created. **onStart()** This callback is called when the activity becomes visible to the user.	1M 1M 6M

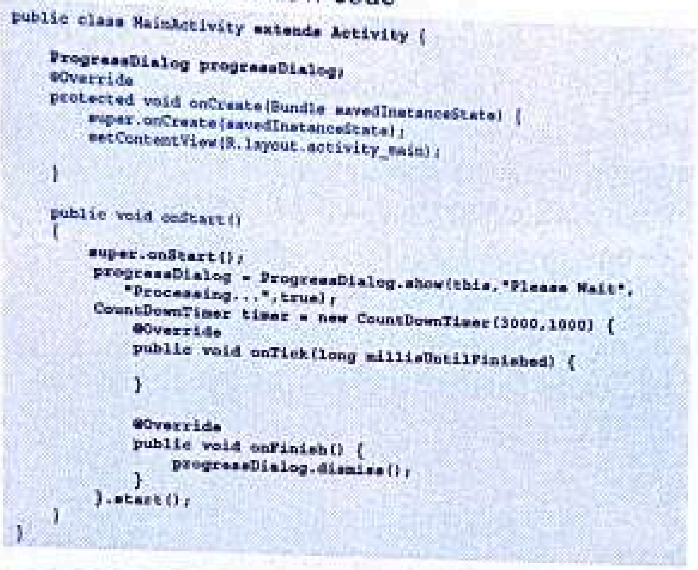
		onResume() This is called when the user starts interacting with the application. **onPause()** The paused activity does not receive user input and cannot execute any code and called when the current activity is being paused and the previous activity is being resumed. **onStop()** This callback is called when the activity is no longer visible. **onDestroy()** This callback is called before the activity is destroyed by the system. **onRestart()** This callback is called when the activity restarts after stopping it. Explanation of above functions	
1	b.	This process of taking users from one application to another is achieved by passing the Intent to the system. Intents, in general, are used for navigating among various activities within the same application, but note, is not limited to one single application, i.e., they can be utilized from moving from one application to another as well. Below are some applications of Intents: • Sending the User to Another App • Getting a Result from an Activity • Allowing Other Apps to Start Your Activity i) Intent i = new Intent(this , ActivityTwo.class); With explanation ii) Intent i = new Intent(getApplicationContext(), ActivityTwo.class); startActivity(i); with explanation	1M 3M 3M
2	a	i) Pass Data Using an Intent Object public class MainActivity extends Activity { @Override	

```
            }
        }

    }
```

With explanation

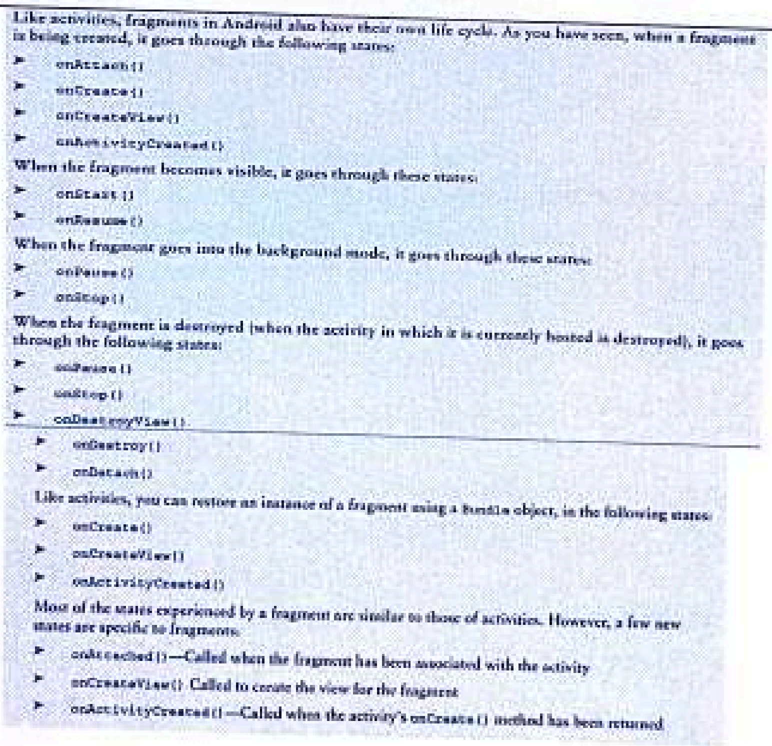
ii) display a progress dialog

Explanation with below code



```
public class MainActivity extends Activity {

    ProgressDialog progressDialog;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

    }

    public void onStart()
    {
        super.onStart();
        progressDialog = ProgressDialog.show(this,"Please Wait",
            "Processing...",true);
        CountDownTimer timer = new CountDownTimer(3000,1000) {
            @Override
            public void onTick(long millisUntilFinished) {

            }

            @Override
            public void onFinish() {
                progressDialog.dismiss();
            }
        }.start();
    }
}
```

2 b



Like activities, fragments in Android also have their own life cycle. As you have seen, when a fragment is being created, it goes through the following states:

- onAttach()
- onCreate()
- onCreateView()
- onActivityCreated()

When the fragment becomes visible, it goes through these states:

- onStart()
- onResume()

When the fragment goes into the background mode, it goes through these states:

- onPause()
- onStop()

When the fragment is destroyed (when the activity in which it is currently hosted is destroyed), it goes through the following states:

- onPause()
- onStop()
- onDestroyView()
- onDestroy()
- onDetach()

Like activities, you can restore an instance of a fragment using a Bundle object, in the following states:

- onCreate()
- onCreateView()
- onActivityCreated()

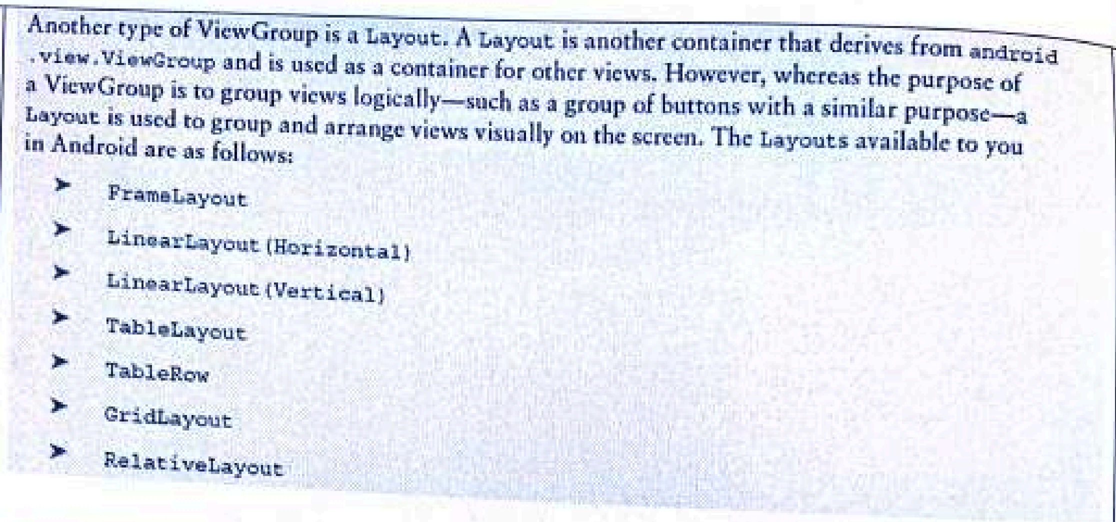
Most of the states experienced by a fragment are similar to those of activities. However, a few new states are specific to fragments:

- onAttached()—Called when the fragment has been associated with the activity
- onCreateView() Called to create the view for the fragment
- onActivityCreated()—Called when the activity's onCreate() method has been returned

With explanation of above methods

7M

```
      public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        LinearLayout parentContainer = new LinearLayout(this);
        parentContainer.setLayoutParams(new
    LayoutParams(LayoutParams.MATCH_PARENT,
            LayoutParams.MATCH_PARENT));
        parentContainer.setOrientation(LinearLayout.VERTICAL);

        Button button = new Button(this);
        button.setText("Open");
        parentContainer.addView(button);                              4M
        button.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View view) {
            Intent i = new Intent("com.java2s.SecondActivity");
            i.putExtra("str1", "This is a string");
            i.putExtra("age1", 25);

            Bundle extras = new Bundle();
            extras.putString("str2", "This is another string");
            extras.putInt("age2", 35);

            i.putExtras(extras);

            startActivityForResult(i, 1);

          }
        });

        setContentView(parentContainer);
    }
    public void onActivityResult(int requestCode, int resultCode, Intent
  data)
    {
      if (requestCode == 1 && resultCode == RESULT_OK) {
        Toast.makeText (this, Integer.toString(
                 data.getIntExtra("age3", 0)),
            Toast.LENGTH_SHORT).show();                               4M
        Toast.makeText (this, data.getData().toString(),
            Toast.LENGTH_SHORT).show();
            fab.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Intent intent = new Intent(getContext(),
AddQuestionActivity.class);
                    intent.putExtra("category", category);
                    startActivity(intent);
                }
            });
        } else {
            emptyTextView.setVisibility(View.GONE);
            fab.setVisibility(View.GONE);
            playQuizAdapter.setValues(questions);
        }
    }
    @Override
    public void showScore() {
        //Create an alert dialog builder
        AlertDialog.Builder builder = new
AlertDialog.Builder(getContext());
        //Set title value
        builder.setPositiveButton("Play Again", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                startActivity(getActivity().getIntent());
            }
        })
                .setNeutralButton("Exit", new
DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        startActivity(new Intent(getContext(),
CategoriesActivity.class));
                    }
                });
        //Get custom view
        LayoutInflater inflater = getActivity().getLayoutInflater();
        final View dialogView = inflater.inflate(R.layout.score_dialog, null);
        builder.setView(dialogView);
```

<table>
<tr>
<td>3</td>
<td>a</td>
<td>


Another type of ViewGroup is a Layout. A Layout is another container that derives from android.view.ViewGroup and is used as a container for other views. However, whereas the purpose of a ViewGroup is to group views logically—such as a group of buttons with a similar purpose—a Layout is used to group and arrange views visually on the screen. The Layouts available to you in Android are as follows:

<ul>
<li>FrameLayout</li>
<li>LinearLayout (Horizontal)</li>
<li>LinearLayout (Vertical)</li>
<li>TableLayout</li>
<li>TableRow</li>
<li>GridLayout</li>
<li>RelativeLayout</li>
</ul>

Justification of above layouts
</td>
<td>1+7=8M</td>
</tr>
<tr>
<td>3</td>
<td>b</td>
<td>
<pre>
@Override
  public void onActivityCreated(@Nullable Bundle savedInstanceState)
{
     super.onActivityCreated(savedInstanceState);
     category =
getActivity().getIntent().getExtras().getString("category");
     fab = getActivity().findViewById(R.id.fab);
     submitButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
           //This checks the array of set integer if it contains 0
           // e.g 0 means radiobutton(option) not selected
           boolean isAllQuestionsAnswered =
playQuizAdapter.answersQuestions.contains(0);
           if (isAllQuestionsAnswered) {
              Toast.makeText(getContext(), "You cannot leave any blank
question", Toast.LENGTH_SHORT).show();
           } else {
              mPresenter.calculateScore();
           }
        }
     });
  }
  @Override
  public void showQuestions(List<Question> questions) {
     if (questions.isEmpty()) {
        showEmptyMessage();
</pre>
</td>
<td>7M</td>
</tr>
</table>

```
        final TextView correctView =
dialogView.findViewById(R.id.correct);
        final TextView incorrectView =
dialogView.findViewById(R.id.incorrect);
        correctView.setText(String.format(Locale.ENGLISH, "%s: %d",
"Correct", playQuizAdapter.getCorrectScore()));
        incorrectView.setText(String.format(Locale.ENGLISH, "%s: %d",
"Incorrect", playQuizAdapter.getIncorrectScore()));
        alertDialog = builder.create();
        alertDialog.show();
      }
```

a

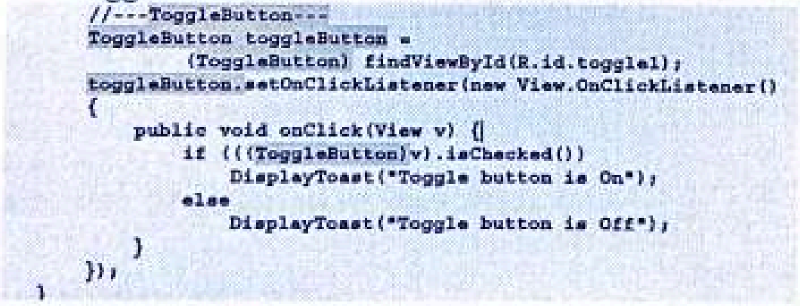
i) Checkbox

Android CheckBox is a type of two state button either checked or unchecked. <CheckBox

```
        android:id="@+id/checkBox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="144dp"
        android:layout_marginTop="68dp"
        android:text="Pizza"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
findViewById<CheckBox>(R.id.checkbox_meat)
   .setOnCheckedChangeListener { buttonView, isChecked ->
      Log.d("CHECKBOXES", "Meat is checked: $isChecked");
   }

findViewById<CheckBox>(R.id.checkbox_cheese)
   .setOnCheckedChangeListener { buttonView, isChecked ->
      Log.d("CHECKBOXES", "Cheese is checked: $isChecked");
   }
```

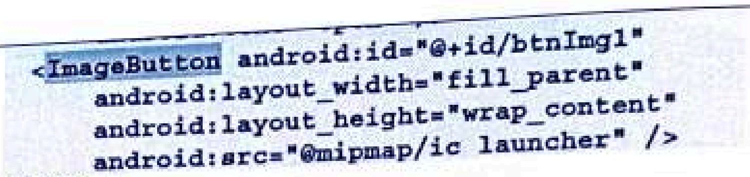
2*4=8M

ii) ToggleButton



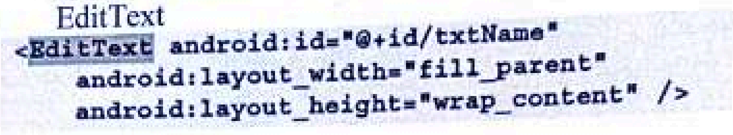
```
    //---ToggleButton---
    ToggleButton toggleButton =
            (ToggleButton) findViewById(R.id.toggle1);
    toggleButton.setOnClickListener(new View.OnClickListener()
    {
        public void onClick(View v) {
            if (((ToggleButton)v).isChecked())
                DisplayToast("Toggle button is On");
            else
                DisplayToast("Toggle button is Off");
        }
    });
}
```

iii) ImageButton



```
<ImageButton android:id="@+id/btnImg1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:src="@mipmap/ic launcher" />
```

iv) EditText



```
<EditText android:id="@+id/txtName"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
```

4 b

Create:

```
public class SqliteManager extends SQLiteOpenHelper {
public static final String DATABASE_NAME = "abhiandroid.db";
public static final int version = 1;

public SqliteManager(Context context) {
  super(context, DATABASE_NAME, null, version);
 }

@Override
public void onCreate(SQLiteDatabase sqLiteDatabase) {
  String dbQuery = "CREATE TABLE Items (id INTEGER PRIMARY KEY
  AUTOINCREMENT,name TEXT, description TEXT)";
  sqLiteDatabase.execSQL(dbQuery);
 }

@Override
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int oldVersion, int
newVersion) {
 }
}
```

7M

Insert:

```
public void insertItem(Item item) {
 String query = "INSERT INTO " + ItemTable.NAME + " VALUES (0,?,?)";
 SQLiteDatabase db = getWritableDatabase();
 db.execSQL(query, new String[]{item.name, item.description});
 db.close();
}
```

Update:

```
public void updateItem(Item item) {
 SQLiteDatabase db = getWritableDatabase();
 ContentValues contentValues = new ContentValues();
 contentValues.put("id", item.id);
 contentValues.put("name", item.name);
 contentValues.put("description", item.description);
 String whereClause = "id=?";
 String whereArgs[] = {item.id.toString()};
 db.update("Items", contentValues, whereClause, whereArgs);
}

Delete:
public void deleteItem(Item item) {
SQLiteDatabase db = getWritableDatabase();
String whereClause = "id=?";
String whereArgs[] = {item.id.toString()};
db.delete("Items", whereClause, whereArgs);
}

Read:
public ArrayList<Item> readAllItems() {
ArrayList<Item> items = new ArrayList<>();
SQLiteDatabase db = getReadableDatabase();
//see above point 2 function
Cursor cursor = db.query("Items"
, null// columns - null will give all
, null// selection
, null// selection arguments
, null// groupBy
, null// having
, null// no need or order by for now;
if (cursor != null) {
  while (cursor.moveToNext()) {
  // move the cursor to next row if there is any to read it's data
          Item item = readItem(cursor);
          items.add(item);
    }
  }
return items;
}

private Item readItem(Cursor cursor) {
 Item item = new Item();
 item.id = cursor.getInt(cursor.getColumnIndex(ItemTable.COL_ID));
 item.name = cursor.getString(cursor.getColumnIndex(ItemTable.COL_NAME));
 item.description =
cursor.getString(cursor.getColumnIndex(ItemTable.COL_DESCRIPTION));
 return item;
}
```

Signature of Faculty

Signature of HOD

ALVA'S
Education Foundation®

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY
(Unit of Alva's Education Foundation (R), Moodbidri)
Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi. Recognized by Government of Karnataka.
A+, Accredited by NAAC
Shobhavana Campus, MIJAR-574225, Moodbidri, D.K., Karnataka
Ph: 08258-262725; Mob:722262724,7026262725,mail:principalaiet08@gmail.com

Rubrics for evaluation of Seminar work in Java for Moblile Applications (18AI63).

Sl No.	Particulars	Marks
1.	Topic related information	10
2.	Presentation	10
3.	Report	10
Total		30

Reduce to 10marks later after evaluation.

Name	USN	Departmer	Semester/	Java for Mi	Java for Mi	Java for Mi	Java for Mi	Java for Mi	Mobile Ap	Mobile Ap	Mobile Ap	Mobile Ap	Mobile Ap
ARAV HANSHIK	4AL20AI005	AIML	6th Sem	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /
Abdullah	4AL20AI001	AIML	6th Sem	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /
AMAN KHADIRSA	4AL20AI002	AIML	6th Sem	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /
Ankit Chavan	4AL20AI003	AIML	6th Sem	Adequate /	Adequate /	Adequate /	High Abilit	High Abilit	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /
Anush L Poojary	4AL20AI004	AIML	6th Sem	High Abilit	Adequate /	Adequate /	High Abilit	High Abilit	Adequate /	Adequate /	Adequate /	Adequate /	Adequate /
Awez ahamed	4al20ai007	AIML	6th Sem	High Abilit	High Abilit	Adequate /	High Abilit	High Abilit	Adequate /	Adequate /	Adequate /	Adequate /	High Ability
B R SUHAAG	4AL20AI008	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Bhavana	4AL20AI016	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Bhoomika	4AL20AI009	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Chirag G	4AL20AI010	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Daksh Uppoor	4al20ai011	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Deekshith R	4AL20AI012	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Deltan Lobo	4AL20AI013	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Divith R Rao	4AL20AI014	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Goutham J S	4AL20AI015	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Harsha k	4al20ai017	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
JESWIN	4AL20AI018	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Karan Kumar	4AL20AI019	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Keerthana K	4AL20AI020	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Keerthanraj M D S	4AL20AI021	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
M ASHOK KUMAR	4al20ai022	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Malini K A	4AL20AI023	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Mohammed Ama	4AL20AI025	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
NIKHIL G B	4AL20AI026	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
NITIN HEMA RAJ	4AL20AI027	AIML	6th Sem	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Prajwal p	4al20ai028	AIML	6th Sem	High Abilit	High Abilit	High Abilit	More than	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Pranjal	4AL20AI029	AIML	6th Sem	High Abilit	High Abilit	High Abilit	More than	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Prasanna Narayar	4AL20AI030	AIML	6th Sem	High Abilit	More than	High Abilit	More than	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Pratham P	4AL20AI031	AIML	6th Sem	More than	More than	High Abilit	More than	High Abilit	High Abilit	High Abilit	High Abilit	High Abilit	High Ability
Prathik N R	4al20ai032	AIML	6th Sem	More than	More than	More than	More than	High Abilit	More than	High Abilit	High Abilit	High Abilit	High Ability
Prathik P Shetty	4AL20AI033	AIML	6th Sem	More than	More than	More than	More than	More than	More than	High Abilit	More than	High Abilit	High Ability
Preetham	4al20ai034	AIML	6th Sem	More than	More than	More than	More than	More than	More than	High Abilit	More than	High Abilit	More than
Puttaraj C Temba	4al20ai035	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than
Sashreeth ks	4al20ai036	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than
Sathyam A V	4al20ai037	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than
Satyam Pawale	4AL20AI038	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than
Shailesh Rao	4AL20AI039	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than

Shivadeep Us	4al20ai040	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than
Shreyas	4AL20AI041	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than
Shriprasad	4AL20AI042	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than
Siddhanth Shetty	4AL20AI043	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than
Souparnika U S	4AL20AI045	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than
Tarun D R	4AI20AI046	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than
Thejas	4AL20AI047	AIML	6th Sem	More than	More than	More than	More than	More than	More than	More than	More than	More than	More than
ULLAS H U	4AL20AI048	AIML	6th Sem	More than	More than	More than	Some Abilit	More than	More than	More than	More than	More than	Some Abilit
Vishma D	4AL20AI049	AIML	6th Sem	More than	More than	Some Abilit	Some Abilit	More than	Some Abilit	Some Abilit	Some Abilit	Some Abilit	Some Abilit
Ashish PB	4AL20AI006	AIML	6th Sem	Adequate /	More than	More than	More than	Adequate Ability		More than	Adequate /	Adequate /	More than

ALVAS INSTITUTE OF ENGINEERING AND TECHNOLOGY MOODBIDRI

Branch : AI Semester : 6

Sl NO.	USN	18AI63
1	4AL20AI001	22
2	4AL20AI002	28
3	4AL20AI003	26
4	4AL20AI004	34
5	4AL20AI005	35
6	4AL20AI006	20
7	4AL20AI007	30
8	4AL20AI008	20
9	4AL20AI009	40
10	4AL20AI010	24
11	4AL20AI011	20
12	4AL20AI012	36
13	4AL20AI013	36
14	4AL20AI014	20
15	4AL20AI015	38
16	4AL20AI016	36
17	4AL20AI017	25
18	4AL20AI018	36
19	4AL20AI019	30
20	4AL20AI020	36
21	4AL20AI021	25
22	4AL20AI022	33
23	4AL20AI023	33
24	4AL20AI025	35
25	4AL20AI026	29
26	4AL20AI027	32
27	4AL20AI028	35
28	4AL20AI029	20
29	4AL20AI030	35
30	4AL20AI031	24
31	4AL20AI032	20
32	4AL20AI033	22
33	4AL20AI034	40
34	4AL20AI035	26
35	4AL20AI036	25
36	4AL20AI037	28

Sl NO.	USN	18AI63
37	4AL20AI038	33
38	4AL20AI039	28
39	4AL20AI040	36
40	4AL20AI041	34
41	4AL20AI042	33
42	4AL20AI043	32
43	4AL20AI045	33
44	4AL20AI046	29
45	4AL20AI047	34
46	4AL20AI048	32
47	4AL20AI049	35

ALVAS INSTITUTE OF ENGINEERING AND TECHNOLOGY MOODBIDRI

Branch : AI Semester : 6

Sl NO.	USN	18AIL68
1	4AL20AI001	33
2	4AL20AI002	30
3	4AL20AI003	30
4	4AL20AI004	33
5	4AL20AI005	40
6	4AL20AI006	33
7	4AL20AI007	40
8	4AL20AI008	29
9	4AL20AI009	40
10	4AL20AI010	27
11	4AL20AI011	28
12	4AL20AI012	40
13	4AL20AI013	40
14	4AL20AI014	26
15	4AL20AI015	40
16	4AL20AI016	38
17	4AL20AI017	29
18	4AL20AI018	40
19	4AL20AI019	40
20	4AL20AI020	40
21	4AL20AI021	30
22	4AL20AI022	40
23	4AL20AI023	40
24	4AL20AI025	40
25	4AL20AI026	30
26	4AL20AI027	38
27	4AL20AI028	40
28	4AL20AI029	25
29	4AL20AI030	39
30	4AL20AI031	25
31	4AL20AI032	25
32	4AL20AI033	29
33	4AL20AI034	40
34	4AL20AI035	31
35	4AL20AI036	27
36	4AL20AI037	33

Sl NO.	USN	18AIL68
37	4AL20AI038	40
38	4AL20AI039	33
39	4AL20AI040	40
40	4AL20AI041	38
41	4AL20AI042	40
42	4AL20AI043	40
43	4AL20AI045	40
44	4AL20AI046	28
45	4AL20AI047	37
46	4AL20AI048	39
47	4AL20AI049	40

BRANCH: ARTIFICIAL INTELLIGENCE & MACHINE LEARNING 5th SEM 2022-2023

			18AI51				18AI62				18ME653				18AI63				18AI644				18AIL66				18AIL67				18AIL68						
			ML				DIP				SCM				JAVA				FDS				ML LAB				DIP LAB				MAD LAB						
Sl.No	USN	NAME OF THE STUDENT	IA	Ext	TOT		IA	Ext	TOT		IA	Ext	TOT		IA	Ext	TOT		IA	Ext	TOT		IA	Ext	TOT		IA	Ext	TOT		IA	Ext	TOT		Grand Total	Avg-%	RESULT
1	4AL20AI001	ABDULLAH	25	21	46	TC	19	18	37	FAIL	26	27	53	SC	22	9	31	FAIL	24	21	45	TC	37	53	90	FCD	31	50	81	FCD	33	48	81	FCD	464	58.00	FAIL
2	4AL20AI002	AMAN KHADIRSAB KONNUR	32	33	65	FC	22	32	54	SC	36	42	78	FCD	28	31	59	SC	33	44	77	FCD	37	56	93	FCD	35	39	74	FCD	30	48	78	FCD	578	72.25	FCD
3	4AL20AI003	ANKIT MANOHAR CHAVAN	33	41	74	FCD	25	30	55	SC	37	35	72	FCD	26	38	64	FC	33	51	84	FCD	33	55	88	FCD	37	48	85	FCD	30	57	87	FCD	609	76.13	FCD
4	4AL20AI004	ANUSH L POOJARY	30	38	68	FC	23	39	62	FC	34	35	69	FC	36	36	72	FCD	30	35	65	FC	33	55	88	FCD	37	51	88	FCD	33	55	88	FCD	600	75.00	FCD
5	4AL20AI005	ARAV HANSHIK	26	38	64	FC	25	32	57	SC	37	38	75	FCD	35	32	67	FC	39	40	79	FCD	36	55	91	FCD	38	51	89	FCD	40	56	96	FCD	618	77.25	FCD
6	4AL20AI006	ASHISH P B	22	21	43	TC	17	15	32	FAIL	31	31	62	FC	20	14	34	FAIL	19	21	40	TC	33	47	80	FCD	37	42	79	FCD	33	0	33	FAIL	403	50.38	FAIL
7	4AL20AI007	AWEZ AHAMED	30	41	71	FCD	28	32	60	FC	40	42	82	FCD	30	36	66	FC	36	42	78	FCD	35	51	86	FCD	35	53	88	FCD	40	56	96	FCD	627	78.38	FCD
8	4AL20AI008	B R SUHAAG	22	24	46	TC	20	26	46	TC	34	36	70	FCD	20	26	46	TC	25	26	51	SC	32	58	90	FCD	29	39	68	FC	29	54	83	FCD	500	62.50	FC
9	4AL20AI009	BHOOMIKA	40	48	88	FCD	40	47	87	FCD	40	53	93	FCD	40	47	87	FCD	40	48	88	FCD	40	60	100	FCD	40	60	100	FCD	40	60	100	FCD	743	92.88	FCD
10	4AL20AI010	CHIRAG G	22	31	53	SC	18	22	40	TC	31	33	64	FC	24	23	47	TC	22	29	51	SC	33	50	83	FCD	27	35	62	FC	27	45	72	FCD	472	59.00	SC
11	4AL20AI011	DAKSH UPPOOR	22	23	45	TC	19	10	29	FAIL	32	29	61	FC	20	23	43	TC	19	21	40	TC	32	45	77	FCD	28	35	63	FC	28	48	76	FCD	434	54.25	FAIL
12	4AL20AI012	DEEKSHITH R	34	44	78	FCD	28	26	54	SC	36	35	71	FCD	40	41	81	FCD	37	40	77	FCD	37	56	93	FCD	37	47	84	FCD	40	55	95	FCD	633	79.13	FCD
13	4AL20AI013	DELTAN GLERAN LOBO	30	42	72	FCD	29	33	62	FC	35	38	73	FCD	36	46	82	FCD	34	41	75	FCD	37	47	84	FCD	38	50	88	FCD	40	58	98	FCD	634	79.25	FCD
14	4AL20AI014	DIVITH R RAO	22	21	43	TC	18	22	40	TC	31	40	71	FCD	20	36	56	SC	19	27	46	TC	32	47	79	FCD	32	32	64	FC	26	47	73	FCD	472	59.00	SC
15	4AL20AI015	SAMNEKAR	36	32	68	FC	28	30	58	SC	36	32	68	FC	38	31	69	FC	35	36	71	FCD	38	60	98	FCD	38	60	98	FCD	40	59	99	FCD	629	78.63	FCD
16	4AL20AI016	H BHAVANA	36	47	83	FCD	38	38	76	FCD	36	42	78	FCD	37	33	70	FCD	38	50	88	FCD	37	57	94	FCD	37	46	83	FCD	38	58	96	FCD	668	83.50	FCD
17	4AL20AI017	HARSHA K	28	21	49	TC	22	21	43	TC	25	28	53	SC	29	24	53	SC	29	21	50	SC	36	45	81	FCD	37	51	88	FCD	29	48	77	FCD	494	61.75	FC
18	4AL20AI018	JESWIN	34	32	66	FC	27	44	71	FCD	35	42	77	FCD	36	41	77	FCD	37	39	76	FCD	34	57	91	FCD	36	48	84	FCD	40	59	99	FCD	641	80.13	FCD
19	4AL20AI019	KARAN KUMAR	38	28	66	FC	29	27	56	SC	30	32	62	FC	38	37	75	FCD	37	35	72	FCD	40	58	98	FCD	39	56	95	FCD	40	57	97	FCD	621	77.63	FCD
20	4AL20AI020	KEERTHANA K	37	30	67	FC	34	40	74	FCD	34	40	74	FCD	36	46	82	FCD	38	39	77	FCD	37	57	94	FCD	37	50	87	FCD	40	57	97	FCD	652	81.50	FCD
21	4AL20AI021	KEERTHANRAJ M D SHETTY	32	35	67	FC	23	21	44	TC	39	35	74	FCD	25	29	54	SC	30	33	63	FC	35	57	92	FCD	37	47	84	FCD	30	57	87	FCD	565	70.63	FCD
22	4AL20AI022	M ASHOK KUMAR	34	41	75	FCD	28	35	63	FC	38	38	76	FCD	33	38	71	FCD	34	46	80	FCD	34	56	90	FCD	38	56	94	FCD	40	57	97	FCD	646	80.75	FCD
23	4AL20AI023	MALINI K A	32	41	73	FCD	29	33	62	FC	33	36	69	FC	39	46	85	FCD	36	47	83	FCD	34	56	90	FCD	38	53	91	FCD	40	57	97	FCD	650	81.25	FCD
25	4AL20AI025	MOHAMMED AMAN	34	44	78	FCD	30	37	67	FC	38	43	81	FCD	35	39	74	FCD	33	43	76	FCD	35	59	94	FCD	38	50	88	FCD	40	58	98	FCD	656	82.00	FCD
26	4AL20AI026	NIKHIL G B	30	32	62	FC	21	28	49	TC	35	37	72	FCD	29	38	67	FC	35	43	78	FCD	34	51	85	FCD	37	56	93	FCD	30	55	85	FCD	591	73.88	FCD
27	4AL20AI027	NITIN HEMA RAJ	33	35	68	FC	32	32	64	FC	26	35	61	FC	32	35	67	FC	39	30	69	FC	38	58	96	FCD	36	46	82	FCD	38	59	97	FCD	604	75.50	FCD
28	4AL20AI028	PRAJWAL P PAVSEKAR	37	40	77	FCD	32	40	72	FCD	31	26	57	SC	35	42	77	FCD	32	35	67	FC	35	56	91	FCD	37	58	95	FCD	40	57	97	FCD	633	79.13	FCD
29	4AL20AI029	PRANJAL NAIDU	25	33	58	SC	18	30	48	TC	20	21	41	TC	36	36	72	FCD	27	38	65	FC	34	51	85	FCD	34	45	79	FCD	25	59	84	FCD	532	66.50	FC
30	4AL20AI030	PRASANNA NARAYANA P	31	26	57	SC	25	33	58	SC	35	41	76	FCD	35	37	72	FCD	32	40	72	FCD	35	57	92	FCD	38	50	88	FCD	39	56	95	FCD	610	76.25	FCD
31	4AL20AI031	PRATHAM P	27	33	60	FC	19	40	59	SC	24	42	66	FC	34	38	72	FCD	26	40	66	FC	35	56	91	FCD	33	41	74	FCD	25	48	73	FCD	561	70.13	FCD
32	4AL20AI032	PRATHIK N R	26	35	61	FC	19	30	49	TC	35	32	67	FC	20	27	47	TC	25	33	58	SC	34	44	78	FCD	30	38	68	FC	25	48	73	FCD	501	62.63	FC
33	4AL20AI033	PRATHIK PADMANABHA SHETTY	26	33	59	SC	19	21	40	TC	32	23	55	SC	22	21	43	TC	20	32	52	SC	34	44	78	FCD	33	37	70	FCD	29	56	85	FCD	482	60.25	FC
34	4AL20AI034	PREETHAM	40	48	88	FCD	40	42	82	FCD	40	54	94	FCD	40	47	87	FCD	40	50	90	FCD	40	60	100	FCD	40	57	97	FCD	40	59	99	FCD	737	92.13	FCD
35	4AL20AI035	PUTTARAJ C TEMBADAMANI	27	36	63	FC	23	31	54	SC	35	42	77	FCD	26	38	64	FC	31	37	68	FC	35	43	78	FCD	33	40	73	FCD	31	49	80	FCD	557	69.63	FC
36	4AL20AI036	SASHREETH K S	30	29	59	SC	19	38	57	SC	36	43	79	FCD	25	34	59	SC	26	29	55	SC	33	39	72	FCD	32	39	71	FCD	27	48	75	FCD	527	65.88	FC
37	4AL20AI037	SATHYAM A V	25	35	60	FC	20	24	44	TC	29	35	64	FC	28	37	65	FC	24	30	54	SC	24	55	79	FCD	37	42	79	FCD	33	56	89	FCD	534	66.75	FC
38	4AL20AI038	SATYAM PAWALE	36	39	75	FCD	32	43	75	FCD	40	41	81	FCD	33	38	71	FCD	37	35	72	FCD	40	57	97	FCD	40	60	100	FCD	40	57	97	FCD	668	83.50	FCD
39	4AL20AI039	SHAILESH RAO	33	37	70	FCD	26	28	54	SC	35	45	80	FCD	28	37	65	FC	33	33	66	FC	35	55	90	FCD	36	53	89	FCD	33	50	83	FCD	597	74.63	FCD
40	4AL20AI040	SHIVADEEP U S	33	43	76	FCD	27	38	65	FC	35	44	79	FCD	36	53	89	FCD	37	46	83	FCD	35	57	92	FCD	34	50	84	FCD	40	55	95	FCD	663	82.88	FCD
41	4AL20AI041	SHREYAS	28	39	67	FC	21	29	50	SC	38	39	77	FCD	34	35	69	FC	32	34	66	FC	36	51	87	FCD	35	50	85	FCD	38	49	87	FCD	588	73.50	FCD
42	4AL20AI042	SHRIPRASAD	30	33	63	FC	35	24	59	SC	25	34	59	SC	33	30	63	FC	30	36	66	FC	36	55	91	FCD	36	50	86	FCD	40	56	96	FCD	583	72.88	FCD
43	4AL20AI043	SIDDHANTH C SHETTY	32	24	56	SC	22	27	49	TC	34	28	62	FC	32	36	68	FC	25	24	49	TC	32	56	88	FCD	35	48	83	FCD	40	50	90	FCD	545	68.13	FC
45	4AL20AI045	SOUPARNIKA U S	31	42	73	FCD	31	41	72	FCD	33	43	76	FCD	40	39	79	FCD	37	34	71	FCD	38	56	94	FCD	37	48	85	FCD	40	56	96	FCD	646	80.75	FCD
46	4AL20AI046	TARUN D R	31	38	69	FC	20	31	51	SC	35	33	68	FC	29	33	62	FC	35	27	62	FC	35	52	87	FCD	34	50	84	FCD	28	49	77	FCD	560	70.00	FCD
47	4AL20AI047	THEJAS	37	47	84	FCD	32	35	67	FC	35	48	83	FCD	34	51	85	FCD	34	59	93	FCD	40	57	97	FCD	36	53	89	FCD	37	59	96	FCD	694	86.75	FCD
48	4AL20AI048	ULLAS H U	31	35	66	FC	27	37	64	FC	38	35	73	FCD	32	33	65	FC	35	40	75	FCD	36	56	92	FCD	36	47	83	FCD	39	59	98	FCD	616	77.00	FCD
49	4AL20AI049	VISHMA D	34	38	72	FCD	31	34	65	FC	39	45	84	FCD	35	35	70	FCD	35	50	85	FCD	40	57	97	FCD	37	59	96	FCD	40	51	91	FCD	660	82.50	FCD
		No.of Fails	0				3				0				2				0				0				0				1						
		FACULTY NAME																																			
		FCD	17				8				28				21				24				47				42				46						
		FC	18				11				13				14				11				0				5				0						
		SC	6				14				5				5				7				0				0				0						
		TC	6				11				1				5				5				0				0				0						

ALVA'S
Education Foundation®

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Unit of Alva's Education Foundation (R), Moodbidri)
Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi.
Recognized by Government of Karnataka.
A+, Accredited by NAAC
Shobhavana Campus, MIJAR-574225, Moodbidri, D.K., Karnataka
Ph: 08258-262725; Mob:722262724,7026262725,mail:principalaiet08@gmail.com

Department of Artificial Intelligence and Machine Learning

Slow Learners:

USN	NAME
4AL20AI001	ABDULLAH
4AL20AI006	ASHISH P B
4AL20AI008	B R SUHAAG
4AL20AI010	CHIRAG G
4AL20AI011	DAKSH UPPOOR
4AL20AI014	DIVITH R RAO
4AL20AI029	PRANJAL NAIDU
4AL20AI031	PRATHAM P
4AL20AI032	PRATHIK N R
4AL20AI033	PRATHIK PADMANABHASHETTY
4AL20AI035	PUTTARAJ C TEMBADAMANI

Fast Learners:

USN	NAME
4AL20AI008	BHOOMIKA
4AL20AI016	H BHAVANA
4AL20AI034	PREETHAM
4AL20AI038	SATYAM PAWALE
4AL20AI040	SHIVADEEP U S
4AL20AI042	SHRIPRASAD
4AL20AI047	THEJAS
4AL20AI049	VISHMA D

CLASS COORDINATOR

HOD

```
 public static void main( String args[] )
 {
  Gender s=Gender.FEMALE;
  if(s==Gender.MALE)
                    System.out.println("Both are not equal");


         }
         }
```

program 2: To find Smallest of given number.

```
enum Value {
                   a(10), b(20);

                   int a1;
                   int getValue(){ return a1;}
                   Value(int value)
                   {
                             this.a1=value;
                   }
         }

class Enu
{
 public static void main( String args[] )
 {
  int s=Value.a.getValue();

  if(s<Value.b.getValue())
            System.out.println("a value is small");
         }
         }
```

2) **if else:** The Java if-else statement also tests the condition. It executes the *if block* if condition is true otherwise *else block* is executed.

Syntax: if(condition) //code if condition is true

else

//code if condition is false
}

program 1:

```
enum Gender {
                  MALE, FEMALE, UNKNOWN;
```

CBCS SCHEME

USN □□□□□□□□□□ 18AI63

Sixth Semester B.E. Degree Examination, June/July 2023
JAVA for Mobile Application

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Module-1

1 a. Explain Java enumerations and write a code to demonstrate how varieties of apple can be represented through enumerations. (08 Marks)

b. Explain following methods with suitable code snippet: (i) Values (ii) ValueOf (04 Marks)

c. Explain type wrappers along with its importance and write a Java program to demonstrate how to use a numeric type wrapper to encapsulate a value and then extract that value. (08 Marks)

OR

2 a. Write a Java code that uses reflection to display the annotation associated with a method. Illustrate all methods used in the program. (10 Marks)

b. Write a Java code to demonstrate auto boxing/unboxing occurs in expressions. (10 Marks)

Module-2

3 a. Explain the following collection classes with suitable code snippet:
(i) The ArrayList Class (ii) The LinkedList Class (10 Marks)

b. Explain the collection framework core interfaces. Describe any two methods associated with Collection Interface. (10 Marks)

OR

4 a. Explain below listed methods with respect to algorithm defined inside collection framework
(i) reverseOrder (ii) Shuffle (10 Marks)

b. Write a Java code to demonstrate custom comparator, which implements the compare( ) method for string that operates in reverse of normal. (10 Marks)

Module-3

5 a. Explain the two string methods that returns the first occurance of a character and last occurance of a character. Illustrate same with suitable Java code. (10 Marks)

b. Illustrate how to modify a string using a following methods:
(i) substring( ) (ii) concat( ) (iii) replace( ) (iv) trim( ) (10 Marks)

OR

6 a. With relevant example, explain the following String Buffer methods:
(i) ensureCapacity( ) (ii) setLength( ) (iii) append( ) (iv) insert( ) (10 Marks)

b. Demonstrate how following methods can be used in character extraction:
(i) charAt( ) (ii) getChars( ) (iii) getBytes( ) (iv) toCharArray( ) (10 Marks)

Module-4

7 a. With a neat block diagram, explain the architecture of android. (10 Marks)

b. With suitable code snippet, explain linking activities using intents. (10 Marks)

OR

8 a. What is an activity? With a neat diagram, explain the Activity Life Cycle. Describe all the events associated. **(10 Marks)**

b. Summarize the states, a fragments goes through after its creation. List the different methods that are called when fragment transists from one state to another. **(10 Marks)**

Module-5

9 a. Describe the following layout available in android: (i) Linear layout (ii) Relative layout **(10 Marks)**

b. Describe progress bar view with suitable code snippet. **(10 Marks)**

OR

10 a. Write a code to build mobile application to retrieve contacts from database. **(10 Marks)**

b. Write a Java code to build a Quiz Application by using Radio Group Class. Consider a suitable view for designing the front end. **(10 Marks)**

* * * * *

COURSE OUTCOMES (COs) ASSESSMENT MATRIX

Alva's Institute of Engineering and Technology, Moodbidri

Department of Artificial Intelligence and Machine Learning

Academic Year:	2022-2023
Course Name & Course Code:	Mobile application development laboratory/ 18AIMP6
Faculty Name:	Shrikanth N G

CO Attainment - Direct				CO Attainment Indirect	CO Attainment
Cos	Formative Assessment	Summative Assessment	Total Attainment Direct		
18AIMP68.1	3	3	3.00	3	3
18AIMP68.2	3	3	3.00	3	3
18AIMP68.3	3	3	3.00	3	3
18AIMP68.4	3	3	3.00	3	3
18AIMP68.5	3	3	3.00	3	3
18AIMP68.6					
Average			3.00	3	3

Faculty Name & Signature:	SHRIKANTH N.G.	HOD Signature:	

Attainment Level 1: 50% students rated more than or equal to 50% of maximum marks

Attainment Level 2: 60% students rated more than or equal to 50% of maximum marks

Attainment Level 3: 70% students rated more than or equal to 50% of maximum marks

Note:

Total Attainment Direct = (Weightage*Formative Assessment)+(Weightage*Summative Assessment)

CO Attainment = (Weightage*Total Attainment Direct)+(Weightage*CO Attainment Indirect)

ALVA'S INSTITUTE OF ENGINEERING & TECHNOLOGY

(Unit of Alva's Education Foundation (R), Moodbidri)
Affiliated to Visvesvaraya Technological University, Belagavi & Approved by AICTE, New Delhi.
Recognized by Government of Karnataka.
A+, Accredited by NAAC
Shobhavana Campus, MIJAR-574225, Moodbidri, D.K., Karnataka
Ph: 08258-262725; Mob:722262724,7026262725,mail:principalaiet08@gmail.com

Department of Artificial Intelligence and Machine Learning

Remedial Classes Attendance

Sem: 6th **Course: Java For Mobile Applications** **Section: A**

USN	NAME	3/6/23	10/6/23	17/6/23	24/6/23
4AL20AI001	ABDULLAH	Ab	Ab	Ab	Ab
4AL20AI006	ASHISH P B				
4AL20AI008	B R SUHAAG				
4AL20AI010	CHIRAG G				
4AL20AI011	DAKSH UPPOOR				
4AL20AI014	DIVITH R RAO				
4AL20AI029	PRANJAL NAIDU	Pranjal	Pranjal	Pranjal	Pranjal
4AL20AI031	PRATHAM P				
4AL20AI032	PRATHIK N R				
4AL20AI033	PRATHIK PADMANABHASHETTY				
4AL20AI035	PUTTARAJ C TEMBADAMANI				

Course Coordinator
[SHRIKANTH N.G]

HOD

PROGRAMME OUTCOME & PROGRAMME SPECIFIC OUTCOME ASSESSMENT MATRIX

Alva's Institute of Engineering and Technology, Moodbidri

Department of Artificial Intelligence and Machine Learning

ALVA'S

Academic Year:	2022-2023
Course Name:	obile application development laborato
Course Code:	18AIMP68
Faculty Name:	Shrikanth N G

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
18AIMP68.1	1	2	2	1	2				1		2	2	2	2	1
18AIMP68.2	2	2	2	1	2						2	2	2	2	1
18AIMP68.3	1	2	2	2	2						1	2	2	2	1
18AIMP68.4	2	2	2	2	2						2	2	2	2	2
18AIMP68.5	2	2	2	1	2						2	2	2	2	2
18AIMP68.6															
AVG	1.6	2	2	1.4	2				1		1.8	2	2	2	1.4

Key Words (POs)	Apply Knowledge	Solve Problems	Design/ Development of Solution	Conduct Investigations	Use Modern Tools	Engineer and Society	Environment and Sustainability	Professional Ethics	Individual and Team Work	Communicate Effectively	Project Management and Finance	Life-long Learning	Professional Skills	Problem Solving Skills	Successful Career & Entrepreneurship
18AIMP68 Direct (	1.6	2	2	1.4	2				1		1.8	2	2	2	1.4
18AIMP68 Indirect	1.6	2	2	1.4	2				1		1.8	2	2	2	1.4
Attainment Level	1.6	2	2	1.4	2				1		1.8	2	2	2	1.4

PO Attainment Calculation Direct

COs	CO Attainment Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
18AIMP68.1	3.00	1	2	2	1	2	0	0	0	1	0	2	2	2	2	1
18AIMP68.2	3.00	2	2	2	1	2	0	0	0	0	0	2	2	2	2	1
18AIMP68.3	3.00	1	2	2	2	2	0	0	0	0	0	1	2	2	2	1
18AIMP68.4	3.00	2	2	2	2	2	0	0	0	0	0	2	2	2	2	2
18AIMP68.5	3.00	2	2	2	1	2	0	0	0	0	0	2	2	2	2	2
18AIMP68.6																
	Weighted Sum --->	8	10	10	7	10	0	0	0	1	0	9	10	10	10	7
	Max Weight ------->	8	10	10	7	10	0	0	0	1	0	9	10	10	10	7
PO Attainment in percentage	:-------->	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00	100.00	0.00	100.00	100.00	100.00	100.00	100.00

PO Attained Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
	1.6	2	2	1.4	2				1		1.8	2	2	2	1.4

PO Attainment Calculation Indirect

COs	CO Attainment Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
18AIMP68.1	3	1	2	2	1	2	0	0	0	1	0	2	2	2	2	1
18AIMP68.2	3	2	2	2	1	2	0	0	0	0	0	2	2	2	2	1
18AIMP68.3	3	1	2	2	2	2	0	0	0	0	0	1	2	2	2	1
18AIMP68.4	3	2	2	2	2	2	0	0	0	0	0	2	2	2	2	2
18AIMP68.5	3	2	2	2	1	2	0	0	0	0	0	2	2	2	2	2
18AIMP68.6																
	Weighted Sum --->	8	10	10	7	10	0	0	0	1	0	9	10	10	10	7
	Max Weight ------->	8	10	10	7	10	0	0	0	1	0	9	10	10	10	7
PO Attainment in percentage	:-------->	100.00	100.00	100.00	100.00	100.00	0.00	0.00	0.00	100.00	0.00	100.00	100.00	100.00	100.00	100.00

PO Attained Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
	1.6	2	2	1.4	2				1		1.8	2	2	2	1.4

Faculty Name & Signature:	SHRIKANTH N.G.

HOD Signature:	

COURSE OUTCOMES (COs) ASSESSMENT MATRIX

Alva's Institute of Engineering and Technology, Moodbidri

Department of Artificial Intelligence and Machine Learning

Academic Year:	2022-2023
Course Name & Course Code	JAVA FOR MOBILE APPLICATIONS/18AI63
Faculty Name:	SHRIKANTH N G

CO Attainment - Direct				CO Attainment Indirect	CO Attainment
Cos	Formative Assessment	Summative Assessment	Total Attainment Direct		
18AI63.1	2	3	2.50	3	2.55
18AI63.2	3	3	3.00	3	3
18AI63.3	3	3	3.00	3	3
18AI63.4	3	3	3.00	3	3
18AI63.5	3	3	3.00	3	3
18AI63.6					
Average			2.90	3	2.91

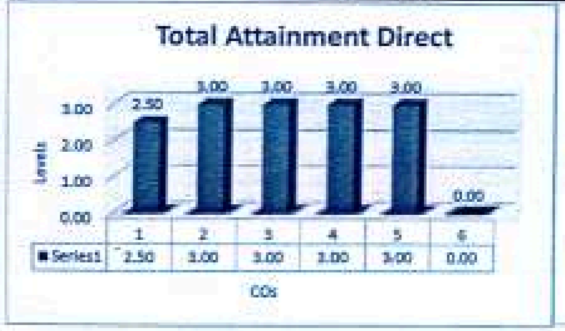
Attainment Level 1: 50% students rated more than or equal to 50% of maximum marks
Attainment Level 2: 60% students rated more than or equal to 50% of maximum marks
Attainment Level 3: 70% students rated more than or equal to 50% of maximum marks

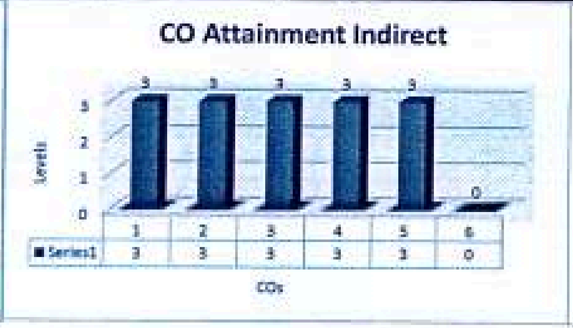
Note:

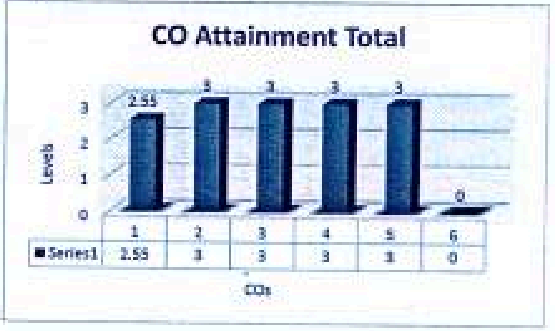
Total Attainment Direct = (Weightage*Formative Assessment)+(Weightage*Summative Assessment) Weightage for Formative Assessment = 50%; Weightage for Summative Assessment = 50%

CO Attainment = (Weightage*Total Attaiinment Direct)+(Weightage*CO Attainment Indirect) Weightage for Total Attainment Direct = 90%; Weightage for CO Attainment Indirect = 10%

Faculty Name & Signature:	SHRIKANTH N G.	HOD Signature:	







PROGRAMME OUTCOME & PROGRAMME SPECIFIC OUTCOME ASSESSMENT MATRIX

Alva's Institute of Engineering and Technology, Moodbidri

Department of Artificial Intelligence & Machine Learning

Academic Year:	2022-2023
Course Name:	JAVA FOR MOBILE APPLICATIONS
Course Code:	18AI63
Faculty Name:	SHRIKANTH N G

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4	PSO5
18AI63.1	2	2	2		2							2	2			2	2
18AI63.2	2	2	2		2							2	2			2	2
18AI63.3	2	2	2		2							2	2			2	2
18AI63.4	2	2	2		2							2	2			2	2
18AI63.5	2	2	2		2							2	2			2	2
18AI63.6																	
AVG	2	2	2		2							2	2			2	2

Key Words (POs)	Apply Knowledge	Solve Problems	Design/ Development of Solution	Conduct Investigations	Use Modern Tools	Engineer and Society	Environment and Sustainability	Professional Ethics	Individual and Team Work	Communicate Effectively	Project Management and Finance	Life-long Learning	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
18AI63 Direct (X)	1.94	1.94	1.94		1.94							1.94	1.94			1.94	1.94
18AI63 Indirect (Y)	2	2	2		2							2	2			2	2
Attainment Level	1.95	1.95	1.95		1.95							1.95	1.95			1.95	1.95

PO Attainment Calculation Direct

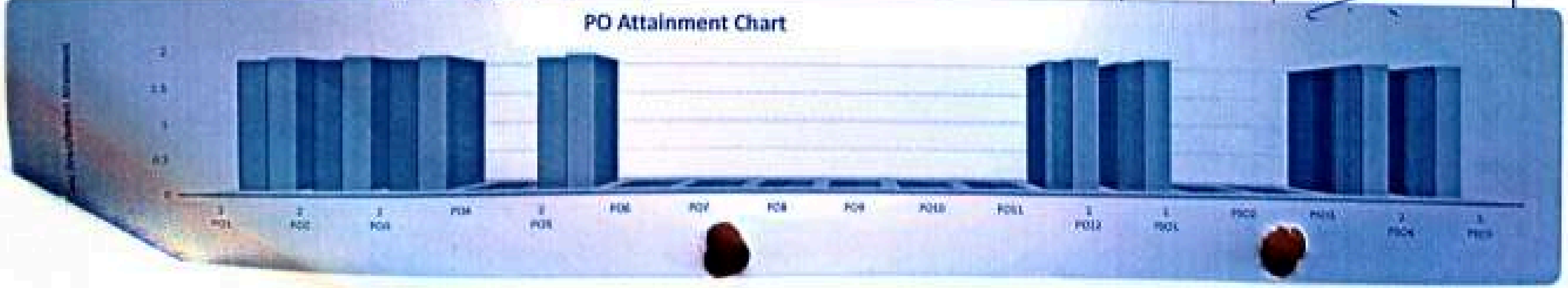
COs	CO Attainment Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4	PSO5
18AI63.1	2.50	1.67	1.67	1.67	0	1.67	0	0	0	0	0	0	1.67	1.67	0	0	1.67	1.67
18AI63.2	3	2	2	2	0	2	0	0	0	0	0	0	2	2	0	0	2	2
18AI63.3	3	2	2	2	0	2	0	0	0	0	0	0	2	2	0	0	2	2
18AI63.4	3	2	2	2	0	2	0	0	0	0	0	0	2	2	0	0	2	2
18AI63.5	3	2	2	2	0	2	0	0	0	0	0	0	2	2	0	0	2	2
18AI63.6																		
	Weighted Sum --->	9.67	9.67	9.67	0	9.67	0	0	0	0	0	0	9.67	9.67	0	0	9.67	9.67
	Max Weight ------>	10	10	10	0	10	0	0	0	0	0	0	10	10	0	0	10	10
PO Attainment in percentage	:------>	96.70	96.70	96.70	0.00	96.70	0.00	0.00	0.00	0.00	0.00	0.00	96.70	96.70	0.00	0.00	96.70	96.70
	PO Attained Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4	PSO5
		1.94	1.94	1.94		1.94							1.94	1.94			1.94	1.94

PO Attainment Calculation Indirect

COs	CO Attainment Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4	PSO5
18AI63.1	3	2	2	2	0	2	0	0	0	0	0	0	2	2	0	0	2	2
18AI63.2	3	2	2	2	0	2	0	0	0	0	0	0	2	2	0	0	2	2
18AI63.3	3	2	2	2	0	2	0	0	0	0	0	0	2	2	0	0	2	2
18AI63.4	3	2	2	2	0	2	0	0	0	0	0	0	2	2	0	0	2	2
18AI63.5	3	2	2	2	0	2	0	0	0	0	0	0	2	2	0	0	2	2
18AI63.6																		
	Weighted Sum --->	10	10	10	0	10	0	0	0	0	0	0	10	10	0	0	10	10
	Max Weight ------>	10	10	10	0	10	0	0	0	0	0	0	10	10	0	0	10	10
PO Attainment in percentage	:------>	100.00	100.00	100.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	100.00	0.00	0.00	100.00	100.00
	PO Attained Grade	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4	PSO5
		2	2	2		2							2	2			2	2

Faculty Name & Signature:	SHRIKANTH N.G.

HOD Signature:	


PO Attainment Chart

Academic Year:	2022-2023
Course Name:	JAVA FOR MOBILE APPLICATIONS
Course Code:	18AI63
Faculty Name:	SHRIKANTH N G

Observations and Action Plan (Direct)

Course	Target	Attainment	Gap	Observation	Action Proposed to bridge the Gap
18AI63	2	2.90	0.90	Target is achieved	Will continue the same

Outcomes on Actions for CAYm1 Observations/Suggestions

SI. No.	Action Taken	Change Observed
1	NA	NA

Faculty Name & Signature:	SHRIKANTH N.G.

HOD Signature:	