A

COURSE FILE

ON

"OBJECT ORIENTED PROGRAMMING THROUGH JAVA"

II B-Tech I Semester

COMPUTER SCIENCE & ENGINEERING

CMR ENGINEERING COLLEGE

KANDLAKOYA (V), MEDCHAL (M), R.R.DIST.

Academic Year 2022-2023

CONTENTS OF COURSE FILE:

1. Department vision & mission
2. List of PEOs, POs, PSOs
3. List of Cos (action verbs as per blooms)
4. Syllabus copy and suggested or reference books
5. Session plan/ lesson plan
6. Session execution log
7. Lecture notes
8. Assignment Questions (samples)
9. Mid exam question papers (samples)
10. Scheme of evaluation
11. Mapping of COs with POs and PSOs
12. COs,POs and PSO s justification
13. Attainment of Cos, Pos and PSOs (Excel sheet)
14. University question papers or question bank.
15. Power point presentations (PPTs)
16. Websites or URLs e- Resources

Submitted By

Desa Uma Vishweshwar

Asst prof,CSE DEPT.

1. DEPARTMENT VISION & MISSION

Vision:

> To produce globally competent and industry-ready graduates in Computer Science & Engineering by imparting quality education with the know-how of cutting-edge technology and holistic personality.

Mission:

1. To offer high-quality education in Computer Science & Engineering in order to build core competence for the graduates by laying a solid foundation in Applied Mathematics and program framework with a focus on concept building.

2. The department promotes excellence in teaching, research, and collaborative activities to prepare graduates for a professional career or higher studies.

3. Creating an intellectual environment for developing logical skills and problem-solving strategies, thus developing, an able and proficient computer engineer to compete in the current global scenario.

2. LIST OF PEOs, POs AND PSOs

2.1 Program Educational Objectives (PEO):

PEO 1: Excel in professional career and higher education by acquiring knowledge of mathematical computing and engineering principles.

PEO 2: To provide an intellectual environment for analyzing and designing computing systems for technical needs.

PEO 3: Exhibit professionalism to adapt current trends using lifelong learning with legal and ethical responsibilities.

PEO 4: To produce responsible graduates with effective communication skills and multidisciplinary practices to serve society and preserve the environment.

2.2. Program Outcomes (POs):

Engineering Graduates will be able to satisfy these NBA graduate attributes:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

2.3 Program Specific Outcomes (PSOs):

PSO1: Professional Skills and Foundations of Software development: Ability to analyze, design and develop applications by adopting the dynamic nature of Software developments.

PSO2: Applications of Computing and Research Ability: Ability to use knowledge in cutting edge technologies in identifying research gaps and to render solutions with innovative ideas.

3. COURSE OUTCOMES

CO1.Solve real world problems using OOPs concepts
CO2.Apply packages, interfaces and handle files
CO3. Demonstrate exceptions and develop multithreading applications
CO4. Design solutions using java collection framework
CO5. Develop applets for Web and GUI based applications.

4. SYLLABUS COPY

UNIT – I

Object-Oriented Thinking- A way of viewing world – Agents and Communities, messages and methods, Responsibilities, Classes and Instances, Class Hierarchies - Inheritance, Method binding, Overriding and Exceptions, Summary of Object -Oriented concepts. Java buzzwords, An Overview of Java, Data types, Variables and Arrays, operators, expressions, control statements, Introducing classes, Methods and Classes, String handling.

Inheritance– Inheritance concept, Inheritance basics, Member access, Constructors, Creating Multilevel hierarchy, super uses, using final with inheritance, Polymorphism - ad hoc polymorphism, pure polymorphism, method overriding, abstract classes, Object class, forms of inheritance- specialization, specification, construction, extension, limitation, combination, benefits of inheritance, costs of inheritance.

UNIT - II

Packages- Defining a Package, CLASSPATH, Access protection, importing packages. Interfaces- defining an interface, implementing interfaces, Nested interfaces, applying interfaces, variables in interfaces and extending interfaces.

Stream based I/O (java.io) – The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, Random access file operations, The Console class, Serialization, Enumerations, auto boxing, generics.

UNIT - III

Exception handling - Fundamentals of exception handling, Exception types, Termination or resumptive models, Uncaught exceptions, using try and catch, multiple catch clauses,

nested try statements, throw, throws and finally, built - in exceptions, creating own exception sub classes.

Multithreading- Differences between thread- based multitasking and process- based multitasking, Java thread model, creating threads, thread priorities, synchronizing threads, inter thread communication.

UNIT - IV

The Collections Framework (java.util)- Collections overview, Collection Interfaces, The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque. Accessing a Collection via an Iterator, Using an Iterator, The For -Each alternative, Map Interfaces and Classes, Comparators, Collection algorithms, Arrays, The Legacy Classes and Interfaces- Dictionary, Hashtable ,Properties, Stack, Vector More Utility classes, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner

UNIT - V

GUI Programming with Swing – Introduction, limitations of AWT, MVC architecture, components, containers. Understanding Layout Managers, Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout.

Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes.

A Simple Swing Application,Applets – Applets and HTML, Security Issues, Applets and Applications, passing parameters to applets. Creating a Swing Applet, Painting in Swing, A Paint example, Exploring Swing Controls- JLabel and Image Icon, JText Field, The Swing Buttons- JButton, JToggle Button, JCheck Box, JRadio Button, JTabbed Pane, JScroll Pane, JList, JCombo Box, Swing Menus, Dialogs.

TEXT BOOKS:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education**.**

REFERENCE BOOKS:

1. An Introduction to programming and OO design using Java, J. Nino and F.A. Hosch, John Wiley & sons
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education. 3. Object Oriented Programming through Java, P. Radha Krishna, University Press.
4. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Oxford Univ. Press.
5. Java Programming and Object-oriented Application Development, R. A. Johnson, Cengage Learning.

5. SESSION PLAN/LESSON PLAN

S.NO	Topic (JNTUH syllabus)	Sub-Topic	NO. OF LECTURES REQUIRED	Suggested Books	Teaching Methods
1	**Unit - 1**				
2	**Object-Oriented Thinking**	Agents and Communities, messages and methods, and Responsibilities	1	T1, T2	BB / PPT
5		Classes and Instances, Class Hierarchies		T1, T2	BB / PPT
7		Inheritance, Method binding	1	T1, T2	BB / PPT
9		Overriding and Exceptions		T1, T2	BB / PPT
10		Summary of Object-Oriented concepts	1	T1, T2	BB / PPT
11		Java buzzwords	1	T1, T2	BB / PPT
12		An Overview of Java		T1, T2	BB / PPT
13		Data types, Variables and Arrays	1	T1, T2	BB / PPT
15		operators, expressions	1	T1, T2	BB / PPT
17		control statements	1	T1, T2	BB / PPT
18		Introducing classes, Methods and Classes	1	T1, T2	BB / PPT
20		String handling	1	T1, T2	BB / PPT
21	**Inheritance**	Inheritance concept, Inheritance basics	1	T1, T2	BB / PPT
23		Member access, Constructors	1	T1, T2	BB / PPT
25		Creating Multilevel hierarchy	1	T1, T2	BB / PPT
26		super uses, using final with inheritance	1	T1, T2	BB / PPT
28		Polymorphism-ad hoc polymorphism	1	T1, T2	BB / PPT
29		pure polymorphism	1	T1, T2	BB / PPT
30		method overriding, abstract classes, Object class	1	T1, T2	BB / PPT

33		forms of inheritance- specialization, specification, construction, extension, limitation, combination	1	T1, T2	BB / PPT
34		benefits of inheritance and costs of inheritance		T1, T2	BB / PPT
36		**Total classes required for UNIT – 1**	17		
37	**UNIT - 2**				
38	**Packages**	Defining a Package	1	T1, T2	BB / PPT
39		CLASSPATH		T1, T2	BB / PPT
40		Access protection		T1, T2	BB / PPT
41		importing packages	1	T1, T2	BB / PPT
42	**Interfaces**	defining an interface		T1, T2	BB / PPT
43		implementing interfaces	1	T1, T2	BB / PPT
44		Nested interfaces		T1, T2	BB / PPT
45		applying interfaces	1	T1, T2	BB / PPT
46		variables in interfaces		T1, T2	BB / PPT
47		extending interfaces		T1, T2	BB / PPT
48	**Stream based I/O**	classes-Byte streams and Character streams	1	T1, T2	BB / PPT
49		Reading console Input and Writing Console Output		T1, T2	BB / PPT
50		File class	1	T1, T2	BB / PPT
51		Reading and writing Files		T1, T2	BB / PPT
52		Random access file operations	1	T1, T2	BB / PPT
53		The Console class		T1, T2	BB / PPT
54		Serialization	1	T1, T2	BB / PPT
55		Enumerations	1	T1, T2	BB / PPT
56		auto boxing		T1, T2	BB / PPT
57		Generics		T1, T2	BB / PPT
58		**Total classes required for UNIT – 2**	9		
59	**UNIT - 3**				
60	**Exception**	Fundamentals of exception handling	1	T1, T2	BB / PPT

		handling			
61		Exception types		T1, T2	BB / PPT
62		Termination or resumptive models	1	T1, T2	BB / PPT
63		Uncaught exceptions		T1, T2	BB / PPT
64		using try and catch	1	T1, T2	BB / PPT
65		multiple catch clauses		T1, T2	BB / PPT
66		nested try statements	1	T1, T2	BB / PPT
67		throw, throws and finally		T1, T2	BB / PPT
68		built- in exceptions	1	T1, T2	BB / PPT
69		creating own exception sub classes	1	T1, T2	BB / PPT
70	**Multithreading**	Differences between thread-based multitasking and process-based multitasking	1	T1, T2	BB / PPT
71		Java thread model		T1, T2	BB / PPT
72		creating threads	1	T1, T2	BB / PPT
73		thread priorities	1	T1, T2	BB / PPT
74		synchronizing threads		T1, T2	BB / PPT
75		inter thread communication	1	T1, T2	BB / PPT
76		**Total classes required for UNIT – 3**	10		
77	**UNIT - 4**				
78	**The Collections Framework**	Collections overview	1	T1, T2	BB / PPT
79		Collection Interfaces		T1, T2	BB / PPT
80		The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque	2	T1, T2	BB / PPT
81		Accessing a Collection via an Iterator	1	T1, T2	BB / PPT
82		Using an Iterator		T1, T2	BB / PPT
83		The For-Each alternative		T1, T2	BB / PPT
84		Map Interfaces and Classes	1	T1, T2	BB / PPT
85		Comparators		T1, T2	BB / PPT

86		Collection algorithms		T1, T2	BB / PPT
87		Arrays		T1, T2	BB / PPT
88		The Legacy Classes and Interfaces-Dictionary, Hashtable ,Properties, Stack, Vector More Utility classes, String Tokenizer, Bit Set, Date, Calendar, Random, Formatter, Scanner	3	T1, T2	BB / PPT
89		**Total classes required for UNIT – 4**	8		
90	**UNIT - 5**				
91	**GUI Programming with Swing**	Introduction and limitations of AWT	1	T1, T2	BB / PPT
92		MVC architecture		T1, T2	BB / PPT
93		Components	1	T1, T2	BB / PPT
94		Containers		T1, T2	BB / PPT
95		Understanding Layout Managers, Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout	1	T1, T2	BB / PPT
96	**Event Handling**	The Delegation event model	1	T1, T2	BB / PPT
97		Events, Event sources, Event Listeners, and Event classes		T1, T2	BB / PPT
98		Handling mouse and keyboard events	1	T1, T2	BB / PPT
99		Adapter classes	1	T1, T2	BB / PPT
100		Inner classes		T1, T2	BB / PPT
101		Anonymous Inner classes		T1, T2	BB / PPT
102	**A Simple Swing Application, Applets**	Applets and HTML	1	T1, T2	BB / PPT
103		Security Issues		T1, T2	BB / PPT
104		Applets and Applications		T1, T2	BB / PPT
105		passing parameters to applets	1	T1, T2	BB / PPT

106		Creating a Swing Applet		T1, T2	BB / PPT
107		Painting in Swing		T1, T2	BB / PPT
108		A Paint example	1	T1, T2	BB / PPT
109		Exploring Swing Controls- JLabel and Image Icon, JText Field		T1, T2	BB / PPT
110	**The Swing Buttons**	JButton		T1, T2	BB / PPT
111		JToggle Button		T1, T2	BB / PPT
112		JCheck Box		T1, T2	BB / PPT
113		JRadio Button		T1, T2	BB / PPT
114		JTabbed Pane	2	T1, T2	BB / PPT
115		JScroll Pane		T1, T2	BB / PPT
116		JList		T1, T2	BB / PPT
117		JCombo Box		T1, T2	BB / PPT
118		Swing Menus		T1, T2	BB / PPT
119		Dialogs		T1, T2	BB / PPT
120		**Total classes required for UNIT – 5**	11		
121		**Total classes required**	**55**		

METHODS OF TEACHING:

M1 : Lecture Method	**M4 : Presentation /PPT**	**M7 : Assignment**
M2 : DemoMethod	**M5 : Lab/Practical**	**M8 : Industry Visit**
M3 : Guest Lecture	**M6 : Tutorial**	**M9 : Project Based**

NOTE:
   1. AnySubjectinaSemesterissupposetobecompletedin55to65periods.
   2. Each Period is of 50minutes.
   3. Each unit duration &completion should be mentioned in the
    Remarks Coloumn.
   4. ListofSuggestedbookscanbemarkedwithCodeslikeT1,T2,R1,R2etc.

INDIVIDUAL TIME TABLE

	I	II	III	IV		V	VI	VII
MON	JAVA-B	JAVA-B						
TUE								
WED						JAVA-D	JAVA-D	
THU							JAVA-B	JAVA-B
FRI	JAVA-D	JAVA-D					JAVA-B	JAVA-B
SAT			JAVA-D	JAVA-D				

6. Session Execution Log: (B Section)

S no	Units	Scheduled started date	Completed date	Remarks
1	I	10.10.2022	21.11.2022	Completed
2	II	24.11.2022	15.12.2022	Completed
3	III	16.12.2022	30.12.2022	Completed
4	IV	02.01.2023	19.01.2023	Completed
5	V	20.01.2023	03.02.2023	Completed

(D section)

S no	Units	Scheduled started date	Completed date	Remarks
1	I	10.10.2022	25.11.2022	Completed
2	II	26.11.2022	3.12.2022	Completed
3	III	16.12.2022	28.12.2022	Completed
4	IV	30.12.2022	20.01.2023	Completed
5	V	21.01.2023	03.02.2023	Completed

7. Lecture Notes – (hand written)

8. ASSIGNMENT QUESTIONS ALONG SAMPLE ASSIGNMENT SCRIPTS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
B. TECH-II -SEM-I MID-I ASSIGNMENT
Subject: OOPs through Java

Answer the following questions

1. Does java support multiple inheritance? Explain nested interface with an example.(CO1)

2. Write a java program to copy contents from source file to destination file. (CO2)

3. What is a package? Explain different ways of importing a package(CO2)

4. What are various forms of inheritance explain with suitable example.(CO1)

5. What is purpose of Abstract classes and differentiate between abstract classes and interface.(CO2)

9.MID EXAM QUESTION PAPER ALONG SAMPLE ANSWER SCRIPTS

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS
(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)
Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
B. TECH-II -SEM-I MIDEXAMINATIONS-I
Subject: OOPs through Java

Time: 10:00 AM to11:30 AM   **Date:** 8-12-2022
Branch: CSE   **M a r k s : 2 5 M**

- -

Note: Question paper contains two parts, Part-A and Part- B.

Part-A is compulsory which carries 10 marks.

Answer all questions in part-A.

Answer anyone full question from each unit.

Each question carries5 marks.

PART-A   5 x 2 M = 10 M

1. What is meant by ad-hoc polymorphism?   (CO1)
2. Explain the differences between abstract class and interface?   (CO2)
3. What are the uses of **final** keyword in java.   (CO1)
4. How to create and access a package?   (CO2)
5. What is an Exception and what are various keywords used to handle exceptions.   (CO3)

PART-B   3 x 5 M = 15 M

6.a) What is an array? How do you declare and access the array in java? Give examples. (CO1)

b) Explain the bitwise and logical operators used in Java.   (CO1)

  (OR)

7. Write aboutTermination and Resumptive models in Exception handling?   (CO3)

8. a) Explain various forms of inheritance with suitable examples   (CO1)

b)Explain the concept of main() overloading with an example?   (CO1)

 (OR)

9. What is the purpose of constructor and Explain the different types of constructors with an example.

                                                                            (CO1)

10.a) Whatare the various way of importing a package? Explain the access protection mechanism in packages?   (CO2)

b) Explain the nested interface concept with a suitable example?   (CO2)

                (OR)

11.a) How do you create and implement an interface in java? (CO2)

b) Write a program to copy the contents of source file to destination file where file names are read from the console. (CO2)

10. Mid-1   SCHEME OF EVALUATION

COURSE: **B.Tech**   YEAR**: II**   SEM: I   A-Y: **2022-23**

NAME OF SUBJECT**:** OOPs Through JAVA PROGRAMMING   MID: **I**   DATE:8-12-2022

PART A

Question Number	Question	subdivision	marks	Total marks
1	What is meant by ad-hoc polymorphism?		2M	2M
2	Explain the differences between abstract class and interface?		2M	2M
3	What are the uses of **final** keyword in java.		2M	2M
4	How to create and access a package?	Creation	1M	2M
		Access modifiers	1M	
5	What is an Exception and what are various keywords used to handle exceptions.	Exception definition	1M	2M
		List the keywords	1M	

PART B

Question Number	Question	subdivision	marks	Total marks
6	a) What is an array? How do you declare and access the array in java? Give examples	Array definition	1M	5M
		Declaration and accessing	1M	
		Example on array	0.5M	
	b) Explain the bitwise and logical operators used in Java.	Bitwise operator	1M	
		Logical operator	1M	
		Example program	0.5M	
7	Write about Termination and Resumptive models in Exception handling?	Description	2.5M	5M
		Example program	2.5 M	
8	Explain various forms of inheritance with suitable examples	Forms of inheritance	2.5M	5M
		Example program	2.5M	
9	What is the purpose of constructor and Explain the different types of constructors with an example	Purpose	1M	5M
		Types	2M	
		Example program	2M	
10	Explain the nested interface concept with a suitable	Concept	2.5M	5M

	example	Example program	2.5M	
11	a)How do you create and implement an interface in java	Description	1M	5M
		Example on interface	1.5M	
	b)Write a program to copy the contents of source file to destination file where file names are read from the console	Program	2.5M	

CMR ENGINEERING COLLEGE
UGC AUTONOMOUS
(Approved by AICTE - New Delhi. Affiliated to JNTUH and Accredited by NAAC & NBA)

Kandlakoya (V), Medchal (M), Medchal - Malkajgiri (D)-501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
B. TECH-II -SEM-I MID-II ASSIGNMENT
Subject: OOPs through Java

Answer the following questions

1. a)Differentiate between multiprocessing and multi threading (CO3)

   b) Explain about inter thread communication with an example. (CO3)

2. Write a program to demonstrate creation of user defined Exception (CO3)

3. Explain about StringTokenizer class with an example(CO4)

4. Explain about Iterator and ListIterator with an example(CO4)

5 a) Explain the life cycle of an Applet with an example(CO5)

   b) Explain about any two LayOut Managers with suitable example.(CO5)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
B. TECH-II -SEM-I   MIDEXAMINATIONS-II
Subject: Object Oriented Programming through Java

Time: 10:00 AM to11:30 AM   **Date: 09-02-2023**
Branch: CSE   **M a r k s : 2 5 M**
- -
Note: Question paper contains two parts, Part-A and Part- B.

Part-A is compulsory which carries 10 marks.

Answer all questions in part-A.

Answer anyone full question from each unit.

Each question carries5 marks.

PART-A   5 x 2 M = 10 M

1.   Write about Checked and Unchecked Exceptions with examples.[CO3]
2.   Differentiate between Multi Processing and Multi Threading.[CO3]
3.   Write various methods available in Enumeration[CO4]
4.   Explain the life cycle of Applet.[CO5]
5.   Explain delegation event model.[CO5]

PART-B   3 x 5 M = 15 M

6) Write a program that demonstrate the creation of user define Exception.[CO3]

(OR)

7) Describe about inter-thread communication with suitable example.[CO3]

8) Explain about the Iterator and ListIterator? [CO4]

(OR)

9) Write briefly about String Tokenizer with an example? [CO4]

10) Explain about any two Layout managers with suitable example [CO5]

(OR)

11) What is the advantage of adaptor classes? Write about any Adaptor class with example program[CO5]

COURSE: **B.Tech**   YEAR**: II**   SEM: I   A-Y: **2022-23**

NAME OF SUBJECT**: OOPs Through** JAVA PROGRAMMING   MID: **II**

DATE:09/02/2023

PART -A

Question Number	Question	subdivision	marks	Total marks
1	Write about Checked and Unchecked Exceptions with examples.		2M	2M
2	Differentiate between Multi Processing and Multi Threading		2M	2M
3	Write various methods available in Enumeration		2M	2M
4	Explain the life cycle of Applet	List of methods	1M	2M
		Diagram	1M	
5	Explain delegation event model	Description	1M	2M
		Diagram	1M	

PART-B

Question Number	Question	subdivision	marks	Total marks
6	Write a program that demonstrate the creation of user define Exception	Program	4M	5M
		Output	1M	
7	Describe about inter-thread communication with suitable example	Description	2M	5M
		Example program	3M	
8	Explain about the Iterator and ListIterator?	Iterator and its program	2.5 M	5M
		ListIterator and its program	2.5 M	
9	Write briefly about String Tokenizer with an example?	Description	2M	5M
		Example program	3M	
10	Explain about any two Layout managers with suitable example	Description	2M	5M
		Example Program	3M	
11	What is the advantage of adaptor classes? Write about any Adaptor class with example program	Advantage	2M	5M
		Example program	3M	

11.Mapping of Course Objectives, Course Outcomes with PEOs and Pos

	Program Outcome(PO):											
	1	2	3	4	5	6	7	8	9	10	11	12
PEOS	x x	x	x									
	x	x				x						
							x				x	
						x			x			

Course Outcomes	Relationship of Course outcomes to Program Outcomes (PO AVG)													
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PS O2
CO1	2		2	2										
CO2		2		2									2	
CO3			3										2	
CO4	3		3	2									2	
CO5	3				2								2	1

12. CO ,POs,PSOs Justification

CO1 WITH PO1: Apply the knowledge of OOPs concepts such as inheritance polymoriphm, abstraction and encapsulation for solving complex engineering problems.Thus overall , the correlation of CO1 with PO1 is moderate.

CO1 WITH PO3: Identify and analyze usage OOPs concepts to solve the problems. Thus overall , the correlation of CO1 with PO3 is moderate

CO1 WITH PO4: By the knowledge on OOPs student will be able to analysis and interpret the data and able to provide the valid conclusions about the problem statement. Thus overall , the correlation of CO1 with PO4 is moderate

CO2 WITH PO2: Identify and analyze usage of interface, packages and handling of files concepts to solve the problems. Thus overall , the correlation of CO2 with PO2 is moderate.

CO2 WITH PO4:with the knowledge on interface, packages and handling of files student will be able to design, analyze and interpret data which will be helpful in investigation of complex problem. Thus overall , the correlation of CO2 with PO4 is moderate

CO2 WITH PSO1: This provides the knowledge and also helps to develop a solution for real-world problems by using Stream based IO,packages and interfaces .So, overall, the correlation of CO2 to PSO1 is moderate.

CO3 WITH PO3:

It contributes only knowledge on thread based multitasking and exception handling which will be helpful in design and develop the solutions for multitasking applications with proper handling of unexpected exceptions in applications. Thus overall , the correlation of CO3 with PO3 is high .

CO3 WITH PSO1 : This provides the knowledge and also helps to develop a solution for real-world problems by using Exception handling and multithreading concepts .So, overall, the correlation of CO3 to PSO1 is moderate.

CO4 WITH PO1: Able to apply the knowledge of Collection framework with includes the legacy and utility classes, to solve the complex engineering problems. Thus overall , the correlation of CO4 with PO1 is high

CO4 WITH PO3:student will be able design solutions and process the needs which are required in solving complex problem by applying the knowledge on collection framework. Thus overall , the correlation of CO4 with PO3 is high

CO4 WITH PO4:with the knowledge on the Framework of collections student is able to design ,analyze and interpret the data for developing the solution. Thus overall , the correlation of CO4 with PO4 is moderate.

CO4 WITH PSO1: Collection framework will make the student to analyze, design and develop application by adopting the creation and accessing method of collections. Thus overall , the correlation of CO4 with PSO1 is moderate.

CO5 WITH PO1: Apply the knowledge of GUI programming ,Event handling, applets and swings in solving complex engineering problems. Thus overall , the correlation of CO5 with PO1 is high

CO5 WITH PO5: Student uses the tools such as Layout Managers, Swings controls and applets in predicting and modeling the complex activities with knowledge on limitations of AWT. Thus overall , the correlation of CO5 with PO5 is moderate.

CO5 WITH PSO1: student is able to analyze design and develop the WEB based applications with good look and feel swing components and controls. Thus overall , the correlation of CO5 with PSO1 is moderate.

CO5 WITH PSO2:student use knowledge of GUI, Event handling, Applet and Swings in rendering solution with innovative ideas. Thus overall , the correlation of CO5 with PSO2 is low.

13.   **ATTAINMENT OF CO's, PO's AND PSO's (EXCELSHEET):**

14. University Question Papers or Question Bank.

R18

Code No: 154BE

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

JAVA PROGRAMMING
(Common to CSE, IT)

Time: 2 Hours   **Max. Marks: 75**

Answer any Five
Questions All Questions
Carry Equal Marks- - -

1. Illustrate constructor overloading concept with an example.
    [15]

2. Define a package. How to import packages? Explain with illustrations.
    [15]

3. How to create multiple threads in a program? Explain with an example.
    [15]

4. Write a program to read content of a file, split the sentences into words and countnumber of occurrences of 'is' in the given text.
    [15]

5. Create a user interface to collect data from customer for opening an account in a bank.Use all possible swing components and layout manager for your interface design. [15]

6. Demonstrate various forms of inheritance with suitable program segments.
    [15]

7. Write a program to copy two files into a target file.
    [15]

8. Explain how to create your own exception in Java program with an example?[15]

R13

B.Tech II Year II Semester Examinations,
December - 2018JAVA PROGRAMMING
(Common to CSE, IT)

Time: 3 Hours   **Max. Marks: 75**

Note: This question paper contains two parts A and B.
Part A is compulsory which carries 25 marks. Answer all questions in Part A.
Part B consists of 5 Units. Answer any one full question from each unit. Each
question carries 10 marks and may have a, b, c as sub questions.

PART- A

(25 Marks)

	What is the significance of Java's byte code?	[2]
b)	List the applications of object oriented programming.	[3]
c)	Differentiate class, abstract class and interface.	[2]
d)	How to create and use a package in Java program?	[3]
e)	How does Java support inter thread communication?	[2]
f)	List any four unchecked exception.	[3]
g)	What is the use of Iterator class?	[2]
h)	Compare byte streams with character streams.	[3]
i)	Give the subclasses of JButton class.	[2]
j)	Differentiate between grid layout and border layout managers.	[3]

PART- B

(50 Marks)

2.a) What are the drawbacks of procedural languages? Explain the need of object
   oriented programming with suitable program.
 b) Discuss the lexical issues of Java.   [5+5]
OR
3.a) What are the primitive data types in Java? Write about type conversions.
 b) What is a constructor? What is its requirement in programming? Explain with
   program.   [5+5]

4.a) With suitable code segments illustrate various uses of 'final' keyword.
 b) Discuss about anonymous inner classes.   [5+5]

OR

5.   What are the benefits of inheritance? Explain the various forms of inheritance with suitable code segments.   [10]

6.a)   With a program illustrate user defined exception handling
  b)   How to handle multiple catch blocks for a nested try block? Explain with an example.   [5+5]

OR

7.a)   Describe how to create a thread with an example.
  b)   Write a program to explain thread priorities

8.   What support is provided by File class for file management? Illustrate with suitable scenarios.   [10]

OR

9.a)   Describe different types of JDBC drivers.
  b)   Explain the random access file operations with a suitable program.   [5+5]

10.a)   What is the role of event listeners in event handling? List the Java event listeners
   b)   Write an applet to display the mouse cursor position in that applet window.[5+5]

OR

11.a)   Discuss various AWT containers with examples.
   b)   Explain about the adapter class with an example.   [5+5]

--ooOoo-

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

B.Tech II Year II Semester Examinations,
December - 2017JAVA PROGRAMMING

(Common to CSE, IT)

Time: 3 Hours   **Max. Marks: 75**

Note: This question paper contains two parts A and B.
Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit.Each question carries 10 marks and may have a, b, c as sub questions.

PART- A

(25 Marks)

   Define data abstraction.   [2]
   b)   What is the size of char data type? Why does it differ from C language?   [3]
   c)   What is the use of anonymous inner class?   [2]
   d)   What is a package? How to define it and access it?   [3]
   e)   Differentiate between error and exception.   [2]
   f)   How to assign priorities to threads?   [3]
   g)   List the functions of Stack class.   [2]
   h)   What is the need of JDBC type 3, type 4 drivers?   [3]
   i)   What are the sources for item event?   [2]
   j)   Give the hierarchy for swing components.   [3]

PART-B

(50 Marks)

2.a)   What feature of Java makes it platform independent and portable?
   b)   Is Java a robust language? Justify your answer.   [5+5]

OR

3.a)   Differentiate between a class and object.
   b)   Demonstrate constructor overloading concept.   [5+5]

4.   What is inheritance? Explain different forms of inheritance with suitable program segments and real world example classes.   [10]

OR

5.a) Differentiate between interface and abstract class.
  b) What is meant by dynamic method dispatch? [5+5]

6.a) Write a program to illustrate the use of multiple catch blocks for a try block.
  b) What are the uses of 'throw' and 'throws' clauses for exception handling? [5+5]

OR

7.a) What is the difference between a thread and a process?
  b) How to achieve synchronization among threads? Write suitable code. [5+5]

8.a) What is a vector? How does it differ from array, list?
  b) Write a program to count number of words in a given sentence. [5+5]

OR

9.a) Write a program to copy the contents of file1 to file 2. Read the names of files as command line arguments.
  b) Write about driver manager class for database connectivity. [5+5]

10.a) Why swing components are preferred over AWT components?
  b) What is an adapter class? What is their role in event handling? [5+5]

OR

11.a) Explain the life cycle of an applet.
  b) What are the various layout managers used in Java? [5+5]

---ooOoo---

15. PPTs AND PRESENTATION

Brief History of Java and Overview of Langauge

java.sun.com/features/1998/05/birthday.html

CS 307 Fundamentals of
Java Basics Computer Science

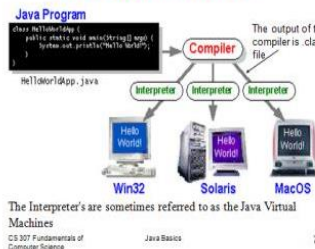### A brief history of Java
"Java, whose original name was Oak, was developed as a part of the Green project at Sun. It was started in December '90 by Patrick Naughton, Mike Sheridan and James Gosling and was chartered to spend time trying to figure out what would be the "next wave" of computing and how we might catch it. They came to the conclusion that at least one of the waves was going to be the convergence of digitally controlled consumer devices and computers."

Applets and Applications
"The team returned to work up a Java technology-based clone of Mosaic they named "WebRunner" (after the movie *Blade Runner*), later to become officially known as the HotJava™ browser. It was 1994. WebRunner was just a demo, but an impressive one: It brought to life, for the first time, animated, moving objects and dynamic executable content inside a Web browser. That had never been done. [At the TED conference.]"

CS 307 Fundamentals of
Java Basics Computer Science 4

How Java Works
- Java's platform independence is achieved by the use of the *Java Virtual Machine*
- A Java program consists of one or more files with a .java extension
  - these are plain old text files
- When a Java program is compiled the .java files are fed to a compiler which produces a .class file for each .java file
- The .class file contains Java bytecode.
- Bytecode is like machine language, but it is intended for the Java Virtual Machine not a specific chip such as a Pentium or PowerPC chip

CS 307 Fundamentals of
Java Basics Computer Science 5

A Picture is Worth…

Java Program

The output of the compiler is .class file.

HelloWorldApp.java

Interpreter Interpreter Interpreter

Win32 Solaris MacOS

The Interpreter's are sometimes referred to as the Java Virtual Machines

CS 307 Fundamentals of
Computer Science Java Basics 7

HelloWorld.java

```
/**
 * A simple program
 */

public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("HELLO CS307!");
    }
}
```

CS 307 Fundamentals of
Java Basics Computer Science 9

Error Types

- Syntax error / Compile errors
  - caught at compile time.
  - compiler did not understand or compiler does not allow
- Runtime error
  - something "Bad" happens at runtime. Java breaks these into Errors and Exceptions
- Logic Error
  - program compiles and runs, but does not do what you intended or want

CS 307 Fundamentals of
Java Basics Computer Science 14

Basic Features

- Data Types
  - primitives
  - classes / objects
- Expressions and operators
- Control Structures
- Arrays
- Methods
- Programming for correctness
  - pre and post conditions
  - assertions

CS 307 Fundamentals of
Java Basics Computer Science 18

Data Types
Primitive Data Types
- byte short int long float double boolean char

```
//dataType identifier;
int x;
int y = 10;
int z, zz;
double a = 12.0;
boolean done = false, prime = true;
char mi = 'D';
```
stick with int for integers, double for real numbers

Classes and Objects
- pre defined or user defined data types consisting of constructors, methods, and fields (constants and fields (variables) which may be primitives or objects.)

CS 307 Fundamentals of
Java Basics Computer Science 19

Java Primitive Data Types

Data Type	Characteristics	Range
byte	8 bit signed integer	-128 to 127
short	16 bit signed integer	-32768 to 32767
int	32 bit signed integer	-2,147,483,648 to 2,147,483,647
long	64 bit signed integer	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	32 bit floating point number	$\pm$1.4E-45 to $\pm$3.4028235E+38
double	64 bit floating point number	$\pm$4.9E-324 to $\pm$1.7976931348623157E+308
boolean	true or false	NA, note Java booleans cannot be converted to or from other types
char	16 bit, Unicode	Unicode character, \u0000 to \uFFFF Can mix with integer types

CS 307 Fundamentals of
Java Basics Computer Science 20

Operators
- Simple Assignment: =
- Arithmetic Operators: +, -, *, /, %(remainder)
  - integer, floating point, and mixed arithmetic and expressions
- Assignment Operators: +=, -=, *=, /=, %=
- increment and decrement operators: ++, --
  - prefix and postfix.
  - avoid use inside expressions.
    - int x = 3;
    - x++;

Casting
- *Casting* is the temporary conversion of a variable from its original data type to some other data type.
  - Like being cast for a part in a play or movie
- With primitive data types if a cast is necessary from a less inclusive data type to a more inclusive data type it is done automatically.
  - int x = 5;
  - double a = 3.5;
  - double b = a * x + a / x;
  - double c = x / 2;
- if a cast is necessary from a more inclusive to a less inclusive data type the class must be done explicitly by the programmer
  - failure to do so results in a compile error.
  - double a = 3.5, b = 2.7;
  - int y = (int) a / (int) b;
  - y = (int) (a / b);
  - y = (int) a / b; //syntax error

CS 307 Fundamentals of
Java Basics Computer Science 22

Primitive Casting

double
float
long
int
short, char
byte

From MORE to LESS

Outer ring is most inclusive data type. Inner ring is least inclusive.

In expressions variables and sub expressions of less inclusive data types are automatically cast to more inclusive.

If trying to place expression that is more inclusive into variable that is less inclusive, explicit cast must be performed.

CS 307 Fundamentals of
Java Basics Computer Science 23

Control Structures
- linear flow of control
  - statements executed in consecutive order
- Decision making with if – else statements

```
if (boolean-expression)
    statement;
if (boolean-expression)
{   statement1;
    statement2;
    statement3;
}
```

A single statement could be replaced by a statement block, braces with 0 or more statements inside

CS 307 Fundamentals of
Java Basics Computer Science 25

for Loops

- for loops
  ```
  for(init-expr;boolean-expr;incr-
  expr)
      statement;
  ```
- init-expr and incr-expr can be more zero or more expressions or statements separated by commas
- statement could be replaced by a statement block
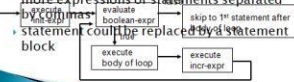
while loops

- while loops
  ```
  while(boolean-expression)
      statement; //or statement block
  ```
- do-while loop part of language
  ```
  do
      statement;
  while(boolean-expression);
  ```
- Again, could use a statement block
- break, continue, and labeled breaks
  - referred to in the Java tutorial as *branching statements*
- keywords to override normal loop logic
- use them judiciously (which means not much)

Arrays in Java

- "Should array indices start at 0 or 1? My compromise of 0.5 was rejected without, I thought, proper consideration. "
  - *S. Kelly-Bootle*
- Java has built in arrays. a.k.a. native arrays
- arrays hold elements of the same type
  - primitive data types or classes
  - space for array must be dynamically allocated with new operator. (Size is any *integer expression*. Due to dynamic allocation does not have to be constant.)

```
public void arrayExamples()
{   int[] intList = new int[10];
    for(int i = 0; i < intList.length; i++)
    {   assert 0 >= i && i < intList.length;
        intList[i] = i * i * i;
    }
    intList[3] = intList[4] * intList[3];
```

Array Details

- arrays must be dynamically allocated
- arrays have a public, final field called *length*
  - built in size field, no separate variable needed
  - don't confuse length (capacity) with elements in use
- elements start with an index of zero, last index is length - 1
- trying to access a non existent element results in an ArrayIndexOutOfBoundsException (AIOBE)

Array Initialization

- Array variables are object variables
- They hold the memory address of an array object
- The array must be dynamically allocated
- All values in the array are initialized (0, 0.0, char 0, false, or null)
- Arrays may be initialized with an initializer list:
  ```
  int[] intList = {2, 3, 5, 7, 11, 13};
  double[] dList = {12.12, 0.12, 45.3};
  String[] sList = {"Olivia", "Kelly", "Isabelle"};
  ```

Array Utilities

- In the `Arrays` class, static methods
- `binarySearch`, `equals`, `fill`, and `sort` methods for arrays of all primitive types (except boolean) and arrays of Objects
  - overloaded versions of these methods for various data types
- In the System class there is an `arraycopy` method to copy elements from a specified part of one array to another
  - can be used for arrays of primitives or arrays of objects

Two Dimensional Arrays

This is our abstract picture of the 2D array and treating it this way is fine.
```
mat[2][1] = 12;
```

Arrays of Multiple Dimension

- because multiple dimensional arrays are treated as arrays of arrays of arrays......multiple dimensional arrays can be *ragged*
  - each row does not have to have the same number of columns
  ```
  int[][] raggedMat = new int[5][];
  for(int i = 0; i < raggedMat.length; i++)
      raggedMat[i] = new int[i + 1];
  ```
  - each row array has its own length field

Methods

- methods are analogous to procedures and functions in other languages
  - local variables, parameters, *instance variables*
  - must be comfortable with variable scope: where is a variable defined?
- methods are the means by which objects are manipulated (objects *state* is changed) – much more on this later
- method header consists of
  - access modifier(**public**, package, protected, **private**)
  - static keyword (optional, class method)
  - return type (void or any data type, primitive or class)
  - method name
  - parameter signature

static methods

- the main method is where a stand alone Java program normally begins execution
- common compile error, trying to call a non static method from a static one

```
public class StaticExample
{   public static void main(String[] args)
    {   //starting point of execution
        System.out.println("In main method");
        method1();
        method2(); //compile error;
    }

    public static void method1()
    {   System.out.println( "method 1");     }

    public void method2()
    {   System.out.println( "method 2");     }
}
```

Assertions

- Assertions have the form

  ```
  assert boolean expression : what to
    output if assertion is false
  ```

- Example

  ```
  if ( (x < 0) || (y < 0) )
  { // we know either x or y is < 0
    assert x < 0 || y < 0 : x + " " + y;
    x += y;
  }
  else
  { // we know both x and y are not less than zero
    assert x >= 0 && y >= 0 : x + " " + y;
    y += x;
  }
  ```

- Use assertion liberally in your code
  - part of style guide

javadoc

- javadoc is a program that takes the comments in Java source code and creates the html documentation pages
- Open up Java source code. (Found in the src.zip file when you download the Java sdk.)
- Basic Format

  ```
  /** Summary sentence for method foo. More details. More
    details.
    pre: list preconditions
    post: list postconditions
    @param x describe what x is
    @param y describe what y is
    @return describe what the method returns
  */
  public int foo(int x, double y)
  ```

- Comments interpreted as html

16. Websites or URLs e- Resources

- http://27.116.21.146/index.php/others/staff-published-articles/cse/cse/category/405-object-oriented-programming-through-java
- https://onlinecourses.nptel.ac.in/noc21_cs03/preview
- https://www.coursera.org/specializations/object-oriented-programming
- http://www.btechsmartclass.com/java/java-index.html
- https://www.tutorialspoint.com/