Government of Karnataka

DEPARTMENT OF COLLEGIATE AND TECHNICAL EDUCATION

Programme	Computer Science and Engineering	**Semester**	IV
Course Code	20CS43P	**Type of Course**	Programme Core
Course Name	Object Oriented Programming and Design with Java	**Contact Hours**	8 hours/week 104 hours/semester
Teaching Scheme	L:T:P :: 3:1:4	**Credits**	6
CIE Marks	60	**SEE Marks**	40

1.Rationale

Object oriented programming paradigm with object-oriented design principles are vital in design and development of today's complex computing solutions. OOD principles provide valuable standards and guidelines to create clean and modular design and avoid code smells. Java being the popular object-oriented programming language that empowers the innovation in this digital world, students will have sound knowledge of object-oriented programming concepts and design principles with java.

2. Course Outcomes: At the end of the course, the student will be able to:

CO-01	Design a solution for a given problem using object-oriented programming concepts and apply all appropriate object-oriented design principles
CO-02	Write and test the code for a designed solution using java OOP concepts.
CO-03	Identify exceptions in the designed or given solution and explain how to resolve them.
CO-04	Demonstrate with an example a java application's connection with a database.

3. Course Content

Week	CO	PO *	Lecture (Knowledge Criteria)	Tutorial (Activity Criteria)	Practice (Performance Criteria)
			3 hours/week	1 hour/week	4 hours/week (2 hours/batch twice in a week)
1	1,2	1, 4	Introduction to Java Brief history; features; java architecture; components: JVM, JRE, JDK; Applications; Java environment setup; Structure of java program; Compilation and execution of java program; Clean coding in java.	**Refer Table 1**	1. Install and Setup java environment 2. Install java editor (Eclipse for Enterprise Java) and configure workspace 3. Execution of first java program 4. Java code execution process
2	1,2	1, 2, 3, 4, 7	Introduction to OOP: Building blocks: class, object, attributes, methods; Class and objects in java;		1. **Code, execute and debug programs** that uses different types of variables and datatypes;

			Variable: Types (local, instance, static); declaration, initialization; comments; 'Data types;		2. Identify and resolve issues in the given code snippet
3	1,2	1, 2, 3, 4	Constructors: rules for defining constructor; types; Destructor; Access modifiers; **this'** keyword; Autoboxing and unboxing; Operators; Expressions; Evaluation of expressions;		**1. Code, execute and debug programs** a. that uses different types of constructors b. for expression evaluation c. to perform autoboxing and unboxing 2. Identify and resolve issues in the given code snippet
4	1,2	1, 2, 3, 4, 7	**Memory allocation in java**; garbage collection: concept, working, types, advantages finalize () method;		1. Install memory monitoring tool and observe how JVM allocates memory 2. Memory allocation explanation through the programs
5	1,2	1, 2, 3, 4	**Conditional and Iterative statements** Decision making: if, if..else, switch Iterative: need of iterative statements; types of loops in java; how to use them; Break and continue statements;		1. Code, execute and debug programs that uses different control statements. 2. Identify and resolve issues in the given code snippet
6	1,2,3	1, 2, 3, 4, 7	**OOP concepts**: **Encapsulation** Concept; What is encapsulation? How to achieve encapsulation in java; Packages; **Single Responsibility Principle**: Intent; Rules; Benefits; example		1. Code, execute and debug programs 2. that uses encapsulation concept. 3. Define class & implement like simple calculator or text processing and check compliance with SRP.
7	1,2	1, 2, 3, 4	**Arrays**: Why arrays? Features, types, Declaration, array creation with new operator, working with arrays; Strings: creation, string methods;		1. Code, execute and debug programs that uses array concept 2. Code, execute and debug programs to perform string manipulation.
8	1,2	1, 2, 3, 4, 7	**OOP concepts**: **Inheritance** Inheritance concept; types; Inheritance in java; Examples; **Open Closed principle:** Intent; Rules; Benefits; example	**Refer Table 1**	1. Code, execute and debug programs that uses inheritance concept 2. Design a class & implement like file parser and check compliance with OCP.

9	1,2	1, 2, 3, 4,7	**OOP concepts: Polymorphism** Polymorphism concept; types: method overloading and overriding; application; polymorphism in java; sufficient examples;		1. Code, execute and debug programs that uses a. static binding b. dynamic binding
10	1,2	1, 2, 3, 4, 7	**OOP concepts: Abstraction** Overview; implementation of abstraction in java: abstract class and interface; Relationship between class and interface; inheritance in interface; Examples to substantiate the understanding of concepts; Eg. File parser; message logger		1. Code, execute and debug programs that uses 2. abstract class to achieve abstraction 3. interface to achieve abstraction 4. Verify whether the given code snippet is correct according to abstraction or not
11	1,2,3	1, 2, 3, 4, 7	**Files and Exception handling** Files and I/O streams: File reader and writer; Exception concept; exceptions in java; classification: checked and unchecked; exception handling in java;		**1. Code, execute and debug programs in java to** a. handles checked and unchecked exceptions b. read the content of the file and write the content to another file 2. Incorporate exception handling in programs/applications developed in previous sessions.
12	1,2,3	1, 2, 3, 4, 7	**Design principle:** **Interface Segregation principle:** Intent; Rules; Benefits; examples; Enums; Overview of java annotations;	**Refer Table 1**	1. Design an interface & implement it like one that builds different types of toys and check compliance with ISP.
13	1,2,3,4	1, 2, 3, 4, 7	**Database Connectivity** Introduction to JDBC; JDBC components; How JDBC works? JDBC connections; Connect java application to database using JDBC;		1. Code, execute and debug programs to connect to database through JDBC and perform basic DB operations
Total in hours			**39**	**13**	**52**

***PO = Program outcome as listed and defined in year 1 curriculum**

Table 1: Suggestive activities for tutorials (the list is only shared as an example and not inclusive of all possible activities for that course. Student and faculty are encouraged to choose activities that are relevant to the topic and the availability of such resources at their institution)

Sl. No	Activity

1	1. Identify various java IDEs and identify differences between them. 2. Compare and contrast Java with Python
2	1. Study and present a. type casting in java b. what are command line arguments in java? c. java keywords and their usage
3	1. Compare and contrast a. method and constructor; b. constructor and destructor
4	1. Study and present how does bytecode work in java.
5	1. Present nesting of conditional and iterative statements considering a use case.
6	Identify advantages and disadvantages of a. Encapsulation. b. Inheritance c. Abstraction d. Polymorphism
7	Study and report a. java Arrays class their methods b. java String class their methods
8	Identify and document how these principles help to avoid code smells. a. SRP b. OCP c. ISP
9	Compare and contrast a. static and dynamic binding and identify usage of each b. abstract class and interface, identify usage of each
10	1. Differentiate error and exception 2. Identify and document system exceptions
11	Study DRY principle, identify the benefits.
12	Identify how OOD principles violations impact the quality of code.
13	Identify java ORM frameworks and their features.
14	Study and find the inclusions in latest java versions.

4. CIE and SEE Assessment Methodologies

Sl. No	**Assessment**	**Test Week**	**Duration** In minutes	**Max marks**	**Conversion**
1.	CIE-1 Written Test	5	80	30	Average of three tests 30
2.	CIE-2 Written Test	9	80	30	
3	CIE-3 Written Test	13	80	30	
4.	CIE-4 Skill Test-Practice	6	180	100	Average of two skill tests reduced to 20
5	CIE-5 Skill Test-Practice	12	180	100	
6	CIE-6 Portfolio continuous evaluation of Activity through Rubrics	1-13		10	10
				Total CIE Marks	60
	Semester End Examination (Practice)		180	100	40
				Total Marks	**100**

5. Format for CIE written Test

Course Name	Object Oriented Programming and Design with Java	Test	I/II/III	Sem	III/IV
Course Code	20CS43P	Duration	80 Min	Marks	30

Note: Answer any one full question from each section. Each full question carries 10 marks.

Section	Assessment Questions	Cognitive Levels	Course Outcome	Marks
I	1			
	2			
II	3			
	4			
III	5			
	6			

Note for the Course coordinator: Each question may have one, two or three subdivisions. Optional questions in each section carry the same weightage of marks, Cognitive level and course outcomes.

6. Rubrics for Assessment of Activity (Qualitative Assessment)

Sl. No.	Dimension	Beginner	Intermediate	Good	Advanced	Expert	Students Score
		2	4	6	8	10	
1		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	8
2		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	6
3		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	2
4		Descriptor	Descriptor	Descriptor	Descriptor	Descriptor	2
	Average Marks= (8+6+2+2)/4=4.5						**5**

Note: Dimension and Descriptor shall be defined by the respective course coordinator as per the activities

7. Reference:

Sl. No.	Description
1	https://docs.oracle.com/javase/tutorial/java/concepts/
2	www.edureka.co
3	Clean Code by Robert C Martin
4	https://www.javabrahman.com/programming-principles/
5	https://medium.com/

8. CIE Skill Test 1 Scheme of Evaluation

SL. No.	Particulars/Dimension	Marks
1	Develop a solution for a given problem using object-oriented programming concepts	20
2	Write program for above given problem using appropriate java OOP concepts.	20
3	Code, execute, test and debug the above program	30
4	Demonstrate the how your program has solved the given problem In the event of, a student fails to get the desired result (with no syntactical errors and least sematic errors), the examiner shall use viva voce to assess the student understanding of OOP concepts and java code execution process.	20
5	Portfolio evaluation based on aggregate of all practice sessions	10

Total Marks	**100**

Note: For CIE skill test 2, SEE scheme of evaluation shall be used.

9. SEE Scheme of Evaluation

SL. No.	Particulars/Dimension	Marks
1	Develop a solution for a given problem using object-oriented programming concepts	20
2	Write program for above given problem using appropriate java OOP concepts.	20
3	Code, execute, test and debug the above program	30
4	Demonstrate how your program has solved the given problem and compliance of your solution with object-oriented design principles. In the event of, a student fails to get the desired result (with no syntactical errors and least sematic errors), the examiner shall use viva voce to assess the student understanding of OOP concepts and OOD principles	20
5	Portfolio evaluation based on aggregate of all practice sessions	10
	Total Marks	**100**

10. Equipment/software list with Specification for a batch of 20 students

Sl. No.	Particulars	Specification	Quantity
1	Computers		20
2	Java 8.0 and above, eclipse		20