U.S.N.										

B.M.S. College of Engineering, Bengaluru-560019

Autonomous Institute Affiliated to VTU

June 2025 Semester End Main Examinations

Programme: B.E. **Semester: III**

Branch: Computer Science and Engineering (Data Science) **Duration: 3 hrs.**

Course Code: 23DS3PCOOJ **Max Marks: 100**

Course: Object Oriented Programming with Java

Instructions: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed.

		UNIT - I	*CO*	*PO*	Marks
1	a)	Demonstrate 3 uses of final keyword in Java with example Program. Can a class be both abstract and final? Justify with example	*CO1*	*PO1*	**08**

	b)	Analyze the erroneous program given below. Identify the errors and discuss. Write the corrected program.
		class Emp
		{
		int eno; static int count;
		Emp( )
		{count++;}
		void seteno(int no)
		{ eno = no;}
		static void display( )
		{
		System.out.println("Emp no:"+eno);
		System.out.println("Emp objects created till now:" +count);
		}
		}
		public class MainClass
		{ public static void main(String args[])
		{ Emp[] employees = new Emp(10);
		for(int i=0; i<employees.length; i++)
		{ employees.seteno(i+1);employees.display( );}
		}
		}
			CO2	*PO2*	**06**

	c)	Explain with neat diagram execution flow of Java program and discuss the importance of Java Bytecode	*CO1*	*PO1*	**06**
		OR			

	2	a)	Consider the following Java program. Identify and explain the errors in the code. Write the correct program for the same.
			class Animal {
			void makeSound();
			final void sleep() {
			System.out.println("Sleeping...");
			}
			}
			interface Mammal {
			void feed();
			//Nested Interface
			interface BabyMammal {
			void crawl();
			}
			}
			class Dog implements Mammal, BabyMammal extends Animal {
			void makeSound() {
			System.out.println("Bark!");
			}
			public void Sleep() {
			System.out.println("Dog Sleeping...");
			}
			public void feed() {
			System.out.println("Feeding...");
			}
			public void crawl() {
			System.out.println("Crawling...");
			}
			}
			public class Main {
			public static void main(String[] args) {
			Dog dog = new Dog();
			dog.makeSound();
			dog.feed();
			dog.crawl();
			dog.sleep();
			}
			}
				CO2	*PO2*	**08**

		b)	Demonstrate the two uses of super keyword with an example program.	*CO1*	*PO1*	**04**
		c)	Write a Java program to create a class PLAYER with instance variables name, matches_played and average. This class has an abstract method cal_average(String,int,int) - String:name, int:matches-played, int:average. Derive two classes BATSMAN and BOWLER from PLAYER. Class BATSMAN has an instance variable runs_scored. Class BOWLER has an instance variable runs_given. Create m BATSMAN objects and n BOWLER objects. Calculate and display the average runs scored by each BATSMAN and average runs given by each BOWLER.	*CO3*	*PO3*	**08**

			UNIT – II			
	3	a)	Illustrate the difference between Method Overloading and Method Overriding with appropriate code snippet	*CO1*	*PO1*	**06**
		b)	Create two user defined package mybank and mycustomer. Package mybank contains a class Customer and a method balance and package mycustomer contains a class MyProfile and a method display. Demonstrate a simple java application program that make use of classes provided by these two packages using an import statement	*CO3*	*PO3*	**08**
		c)	With an example program illustrate defining interface, nested interfaces and implementing interfaces.	*CO1*	*PO1*	**06**
			OR			
	4	a)	Tabulate the various levels of access protection available for packages and their implications. Create two packages and demonstrate the various levels of access protections with appropriate program	*CO1*	*PO1*	**07**
		b)	Analyze the code and rectify the errors in the code. Justify your answer. **package p1;** **public class Protection {** **int n = 1;** **private int n_pri = 2;** **protected int n_pro = 3;** **public int n_pub = 4;** **public Protection() {** **System.out.println("n = " + n);** **System.out.println("n_pri = " + n_pri);** **System.out.println("n_pro = " + n_pro);** **System.out.println("n_pub = " + n_pub);** **}** **}** **class Derived extends Protection {** **Derived() {** **System.out.println("n = " + n);** **System.out.println("n_pro = " + n_pro);** **System.out.println("n_pub = " + n_pub);** **}** **}** **class SamePackage {** **SamePackage() {** **Protection p = new Protection();** **System.out.println("n = " + super.n_pro);** **System.out.println("n_pro = " + p.n_pro);** **System.out.println("n_pub = " + p.n_pub);** **}** **void show(){** **System.out.println("Hello world");** **}** **}** **package p2;** **class Protection2 extends Protection {** **Protection2() {** **System.out.println("n = " + n);** **System.out.println("n_pro = " + n_pro);** **System.out.println("n_pub = " + n_pub);** **}** **}** **class OtherPackage implements Inner {** **OtherPackage() {** **pi=3.24;** **p1.Protection p = new p1.Protection();** **System.out.println("n_pro = " + p.n_pro);** **System.out.println("n_pub = " + p.n_pub);** **}** **void show(){** **System.out.println("Inside show");** **}** **}** **interface Outer{** **double pi=3.14;** **void display();** **}** **interface Inner extends Outer{** **void show();** **}**	*CO2*	*PO2*	**07**
		c)	A person is eligible to be a U.S. Senator who is at least 30 years old and has been a U.S. citizen for at least 9 years. To be house, at least 7 years. Create a method that takes person's age and years of citizenship as arguments and returns either "None", "House", "Senate", or "House and Senate" depending on which elected	*CO3*	*PO3*	**06**

			position(s) they can hold. Implement a Java program using this method to ask the user for their age and years of citizenship and then display them in which legislative position(s) they could serve.			
			UNIT - III			
	5	a)	Write a Java program which creates two threads, one thread displaying "BMS College of Engineering" once for every ten seconds and another displaying "CSE" once for every 2 seconds.	*CO3*	*PO3*	**06**
		b)	Demonstrate the use of isAlive() and join() with an example program	*CO1*	*PO1*	**06**
		c)	Create a class Student which includes instance variables usn, name and marks of three subjects. Accept the input through command line arguments. Raise a user defined exception 'noargs' if no arguments are given in command line. Also raise an exception when negative marks are given.	*CO3*	*PO3*	**08**
			OR			
	6	a)	Develop a multithreaded Java program to create 3 threads. First thread generates random integer for every second, and if the value is even, second thread computes the square of the number and prints. If the value is odd third thread print the value of cube of the number.	*CO3*	*PO3*	**08**
		b)	Demonstrate the use of throws clause and finally block in exception handling with an example program.	*CO1*	*PO1*	**07**

		c)	Analyze the following program to predict the output.
			class NestedTry {
			public static void main(String args[]) {
			try {
			int a[] = { 1, 2, 3, 4, 5 };
			try {
			int x = a[2] / 0;
			} catch (ArithmeticException e2) {
			System.out.println("division by zero is not possible");
			System.out.println(a[5]);
			} finally {
			System.out.println("Outside Inner Nested try block");
			}
			} catch (ArrayIndexOutOfBoundsException e1) {
			System.out.println("ArrayIndexOutOfBoundsException");
			} finally {
			System.out.println("Outside Outer Nested Try Block");
			}
			System.out.println("Element at such index does not exist");
			}
			}
				CO2	*PO2*	**05**

			UNIT - IV			
	7	a)	Write a Java program that reads a collection of words and prints words in alphabetical order.	*CO3*	*PO3*	**07**
		b)	Write a program to accept a file name as command line parameters. Check whether File exists in C:\Java directory. If it exists display its name and size, else, display message that it does not exist.	*CO3*	*PO3*	**07**

		c)	Given the line of code:
			StringBuffer sb = new StringBuffer("Bengaluru");
			Write the output, when the following string methods are called:
			i) char c=sb.charAt(7); System.out.println(c);
			ii) char[] buf = new char[7];
			sb.getChars(2, 7,buf,0);System.out.println(buf);
			iii) sb.replace(0, 4, "Mang"); System.out.println(sb);
			iv) sb.delete(3, 4);System.out.println(sb);
				CO2	*PO2*	**06**

			OR			
	8	a)	Write a java program to count the number of characters, words and lines in a given file, filename is provided as an argument by the user.	*CO3*	*PO3*	**07**
		b)	Write java code snippets using string methods for following operations i)Replace all occurrences of "Edureka" to "Brainforce". Therefore, the output would be "Hey, welcome to Brainforce". ii)Compares the two strings(Str1=DATA SCIENCE, Str2=data science) on the basis of content of the string. iii)Check if string Str1 ends with the suffix - SCIENCE.	*CO3*	*PO3*	**06**
		c)	Write a Java program using String operations to accept a list of full names from the user and computes the initials for those full names. Display the initials to the user. (Eg: “Puneeth Rajkumar” – initial is PR)	*CO3*	*PO3*	**07**
			UNIT - V			
	9	a)	Explain Generics. Demonstrate generics with two parameters with an example program.	*CO1*	*PO1*	**07**
		b)	Explain ArrayDequeue class. Write Java program to demonstrate the ArrayDeque class in Java having following operations in the same order. 1. Insert 10,20,30,40,50 and display the dequeue 2. Empty the dequeue 3. Insert the element at the head 4. Insert the element at the tail and display the dequeue Write the output for the same.	*CO3*	*PO3*	**07**

		c)	Compare type wrappers with primitive types in Java, discussing advantages and disadvantages in terms of memory usage, performance, and functionality.	*CO1*	*PO1*	**06**
			OR			
	10	a)	Demonstrate and discuss the concepts of Autoboxing and Unboxing with a Java program	*CO1*	*PO1*	**06**
		b)	Write a program to illustrate various methods of hashset.	*CO1*	*PO1*	**07**
		c)	Compare ArrayList and LinkedList regarding their data structure, performance, and preferred usage scenarios, focusing on insertion, deletion, and element access operations.	*CO1*	*PO1*	**07**

****