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 **Duration: 3 hrs.**

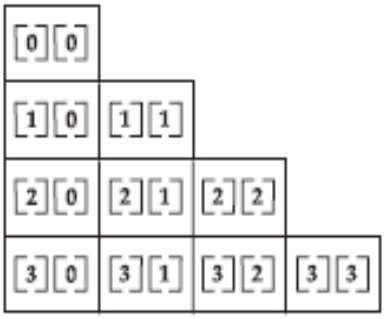
Course Code: 22CS3PCOOJ **Max Marks: 100**

Course: Object Oriented Java Programming

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

Important Note: Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

		UNIT - I	CO	PO	Marks
1	a)	Explain Type conversion and Type casting and demonstrate with an appropriate program	CO1	PO1	**6**

	b)	Complete the code snippet to create a 2d array in the format given here.
		class TwoDA
		{
		public static void main(String args[])
		{
		int twoD[][] =
		……
		……
		}
		}
		 
		
			CO2	PO2	**6**

	c)	Create a class Distance with members feet and inches. Include methods to set the values, print the values set, add the given two distances considering the fact that 12 inches = 1 foot and return back the resultant distance object to the main method.	CO3	PO3	**8**
		OR			
2	a)	Discuss on how Java is Robust and Architecture-Neutral.	CO1	PO1	**6**
	b)	Analyze the errors in the given program. Write the corrected output. Underline the places where errors are corrected	CO2	PO2	**6**

```
class Box {
double width; double height; double depth;
void volume ( )
{
 System.out.print("Volume is ");
 System.out.println(width * height * depth);
 return (width * height * depth);
 } }

class BoxDemo3
{
   public static void main (String args[])
   {
     Box mybox1 = new Box();
     Box mybox2;
     Box mybox1 = new Box (10,10,10);
      mybox1.width = 10;  mybox1.height = 20;  mybox1.depth = 15;
       mybox2.width = 3;   mybox2.height = 6; mybox2.depth = 9;
       mybox1.volume( );
       mybox2.volume( );
       mybox3.volume( );
      }}
```

		c)	Develop a Java program to create a class Student with members- usn, name, marks in three courses, sum and average. Create an array of n Student objects. Accept the details and calculate the sum and average. Include the functionality to print the details of the topper.	*CO3*	*PO3*	**8**
			UNIT - II			
3		a)	Explain dynamic method dispatch with an example program.	*CO1*	*PO1*	**6**

		b)	Analyze the output given below and write a Java program which displays the same on execution. Use constructor(s) with inheritance.
			`Super1 created`
			`Inside Super1 default constructor`
			`Sub1 created`
			`Inside Super1 default constructor`
			`Inside Sub1 default constructor`
			`Sub11 created`
			`Inside Super1 default constructor`
			`Inside Sub1 default constructor`
			`Inside Sub11 default constructor`
				CO2	*PO2*	**6**

		c)	Create a class Student with members- usn, name, age, dept and sem. Include methods to set and print the values. Derive two subclasses- PG_student with member intern_companyname and Research Scholar with member no_publications. Create n objects for each of the classes. Include methods to do the following i) Print the name of the PG_students who have internship in a company of user's choice ii) Print the details of scholars who have 0 publications.	*CO3*	*PO3*	**8**

			OR			
	4	a)	Demonstrate the three uses of final keyword.	*CO1*	*PO1*	**6**

		b)	Write the expected output of the program given below.
			class A
			{int value=100; byte b=50;
			void display( ) { b=(byte)(b*4);
			System.out.println("Class A method:value= "+value+" b = "+b);
			}
			void printMsg( )
			{ display( ); } }
			class B extends A
			{ int value=200;
			void display( )
			{ System.out.println("Class B method- value "+value); }
			void printMsg( )
			{ value+=super.value; display( ); super.display( ); } }
			class C
			{ public static void main(String args[])
			{ A obj= new B( ); obj.display( );
			System.out.println(obj.value); obj.printMsg( ); } }
				CO2	*PO2*	**6**

		c)	Create an abstract class **Calculate** which has three double members -say **x**, **y** and **result**. Include a method **calc**. Derive three classes from **Calculate** which performs any three arithmetic operations on the two variables **x** and **y** and assign the result to the variable **result**. Make appropriate declarations and definitions.	*CO3*	*PO3*	**8**
			UNIT - III			
	5	a)	Demonstrate finally keyword with an example program	*CO1*	*PO1*	**6**

		b)	Analyze the given program, find the errors. Write the corrected program. Underline the places where errors are corrected.
			interface MyInterface
			{
			void method1();
			void method2();
			}
			class Demo extends MyInterface
			{
			void method1()
			{
			System.out.println("implementation of method1");
			}
			void method2()
			{
			System.out.println("implementation of method2");
			}
			public static void main (String arg[])
			{
			MyInterface obj = new MyInterface ();
			obj. method1();
			}
			}
				CO2	*PO2*	**6**

		c)	Write a program to create a user defined exception called “Student not found” which accepts a string message. Create another class “Student Manager” in which the student id is checked to be equal to 121. If not equal to 121 the created user defined exception should be thrown. Create a test class to handle the exception appropriately.	*CO3*	*PO3*	**8**
			OR			
	6	a)	Demonstrate interfaces with an example program.	*CO1*	*PO1*	**6**

		b)	Analyse the program and determine the output when the variable a1=0 and a1=1. Justify your answer with appropriate reason.
			`class Exceptionhan {`
			`public static void main (String args[]){`
			`int a1=1;`
			`try {`
			`try {`
			`System.out.println("going to divide");`
			`int b =39/a1;`
			`try {`
			`int a[]=new int[5];`
			`a[5]=4;`
			`} catch (ArithmeticException e) {System.out.println(e);}`
			`} catch (ArrayIndexOutOfBoundsException e) {System.out.println("Array overflow");}`
			`System.out.println("other statement");`
			`} catch (Exception e) {System.out.println("handeled");}`
			`System.out.println("normal flow..");`
			`}`
			`}`
				CO2	*PO2*	**6**

		c)	Create a user defined exception named Reorder. Create a class called Item with members- Id, name, price, quantity, reorder_level. Include methods- i) purchase- which adds items to the existing quantity and ii) sell- which reduces the same. Create two objects of Item and set the values given by the user. Raise the exception Reorder when the quantity of any item goes below reorder_level.	*CO3*	*PO3*	**8**
			UNIT - IV			
	7	a)	Discuss the functionalities and syntax of isAlive() and join() methods with an example program.	*CO1*	*PO1*	**6**

		b)	Modify the given program to get the output shown.
			Program:
			`class A extends Thread { …… }`
			`class Main{`
			`public static void main(String args[]){`
			`A a1=new A();`
			`System.out.println("Name of thread 't':"+ t.getName( ));`
			`}`
			`}`
			Expected Output:
			Name of thread 't': FirstThread
				CO2	*PO2*	**6**

			New name of thread 't': NewThread Thread is running.			
		c)	Develop a Java Program to create three Threads using runnable Interface. Make all threads to execute for five iterations. Set the name of the three threads as “FIRST”, “SECOND”, “THIRD”. Make the second thread to terminate for the 4th Iteration and last thread to sleep for two seconds after two iterations	*CO2*	*PO2*	**8**
			OR			
	8	a)	Explain thread priorities. Include the syntax of the method use to set the priority of threads and explain.	*CO1*	*PO1*	**6**

		b)	Analyze the given Thread code snippet in Thread and add necessary codes to print the expected output:
			class A extends Thread{
			public static void main(String args[]){
			A t=new A();
			System.out.println("Name of thread 't':"+ t.getName());
			}
			}
			Expected Output:
			Name of thread 't':Thread-0
			New name of thread 't': BMSCE
			Thread is running.
				CO2	*PO2*	**6**

		c)	Develop a Java program to print the output in the following order only - [BMS] [College] [of] [Engineering] using threads. Consider each of the words in output to be sent as a separate thread parameter.	*CO3*	*PO3*	**8**
			UNIT - V			
	9	a)	Explain the AWT class hierarchy with a neat diagram.	*CO1*	*PO1*	**6**
		b)	Create a Java program to draw a polygon of five sides and arc from 12 ‘o’ clock to 6 ‘o’ clock.	*CO3*	*PO3*	**6**
		c)	Write a Java program to handle any six mouse events- any four under MouseListener and two under MouseMotionListener.	*CO3*	*PO3*	**8**
			OR			
	10	a)	Write about the delegation event model.	*CO1*	*PO1*	**6**
		b)	Create the child frame window, set the title of the frame and handle window closing event using suitable adapter classes	*CO3*	*PO3*	**6**
		c)	Develop a Java program to create the smiley graphic image given below. 	*CO3*	*PO3*	**8**

*****