CBCS SCHEME

USN | | | | | | | | | | |

BCS306A

Third Semester B.E./B.Tech. Degree Examination, Dec.2023/Jan.2024
Object Oriented Programming with Java

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	Discuss the different data types supported by Java along with the default values and literals.	8	L2	CO1
	b.	Develop a Java program to convert Celsius temperature to Fahrenheit.	6	L3	CO2
	c.	Justify the statement "Compile once and run anywhere" in Java.	6	L2	CO1
		OR			
Q.2	a.	List the various operators supported by Java. Illustrate the working of >> and >>> operators with an example.	8	L2	CO1
	b.	Develop a Java program to add two matrices using command line argument.	10	L3	CO2
	c.	Explain the syntax of declaration of 2D arrays in Java.	2	L2	CO1
		Module – 2			
Q.3	a.	Examine Java Garbage collection mechanism by classifying the 3 generations of Java heap.	6	L2	CO1
	b.	Develop a Java program to find area of rectangle, area of circle and area of triangle using method overloading concept. Call these methods from main method with suitable inputs.	10	L3	CO2
	c.	Interpret the general form of a class with example.	4	L2	CO2
		OR			
Q.4	a.	Outline the following keywords with an example : (i) this (ii) static	6	L2	CO2
	b.	Develop a Java program to create a class called 'Employee' which contains 'name', 'designation', 'empid' and 'basic salary' as instance variables and read ( ) and write ( ) as methods. Using this class, read and write five employee information from main ( ) method.	10	L3	CO2
	c.	Interpret with an example, types of constructions.	4	L2	CO2
		Module – 3			
Q.5	a.	Illustrate the usage of super keyword in Java with suitable example. Also explain the dynamic method dispatch.	10	L2	CO3
	b.	Build a Java program to create an interface Resizable with method resize (int radius) that allow an object to be resized. Create a class circle that implements resizable interface and implements the resize method.	10	L3	CO3
		OR			
Q.6	a.	Compare and contrast method overloading and method overriding with suitable example.	8	L2	CO2

	b.	Define inheritance and list the different types of inheritance in Java.	4	L2	CO3
	c.	Build a Java program to create a class named 'Shape'. Create 3 sub classes namely circle, triangle and square ; each class has 2 methods named draw ( ) and erase ( ). Demonstrate polymorphism concepts by developing suitable methods and main program.	8	L3	CO3
		Module – 4			
Q.7	a.	Examine the various levels of access protections available for packages and their implications with suitable examples.	10	L2	CO4
	b.	Build a Java program for a banking application to throw an exception, where a person tries to withdraw the amount even though he/she has lesser than minimum balance (Create a custom exception)	10	L3	CO4
		OR			
Q.8	a.	Define Exception. Explain Exception handling mechanism provided in Java along with syntax and example.	10	L2	CO4
	b.	Build a Java program to create a package "balance" containing Account Class with displayBalance ( ) method and import this package in another program to access method of Account Class.	10	L3	CO4
		Module – 5			
Q.9	a.	Define a thread. Also discuss the different ways of creating a thread.	6	L2	CO5
	b.	How synchronization can be achieved between threads in Java? Explain with an example.	6	L2	CO5
	c.	Develop a Java program for automatic conversion of wrapper class type into corresponding primitive type that demonstrates unboxing.	8	L3	CO5
		OR			
Q.10	a.	Summarize the type wrappers supported in Java.	6	L2	CO5
	b.	Explain Autoboxing/Unboxing that occurs in expressions and operators.	6	L2	CO5
	c.	Develop a Java program to create a class myThread. Call the base class constructor in this class's constructor using super and start the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently.	8	L3	CO5

* * * * *

Visvesvaraya Technological University

Belagavi, Karnataka - 590 018.

Scheme & Solutions — Signature of Scrutinizer

Subject Title : Object Oriented Programming with Java **Subject Code :** BCS306A

Question Number	Solution	Marks Allocated
1 a)	Data types in Java Primitive: byte, short, int, long; float, double; boolean; char String Reference: Array, class, interface Listing – 2 Explanation – 3 default values – 2 literals – 1	

b)	`public class Main{`
	`public static void main(String[]args){` – Defining class & main method – 1
	`double c, f;`
	`Scanner sc = new Scanner(System.in);` – Reading celsius temp – 2
	`c = sc.nextDouble();`
	`f = 1.8 * c + 32;` – Converting
	`System.out.println("Fah. temp"+f);}}` – Displaying temp – 1	

c)	In Java, the program is converted to bytecode (.class file) which is not native to the processor & interpreted by JVM. So once it is compiled, bytecode is generated which can run anywhere in any machine	6
	OR	
2 a)	Listing operators – 2 Explanation of >> (Rightshift operator) – 3 Explanation of >>> (Rightshift with zero fill) – 3	
b)	passing the order of matrices through command line args – 2 Reading 2 matrices logic – 3 Adding 2 matrices logic – 3 Displaying the resultant matrix – 2	

Subject Title : Object Oriented Programming with Java **Subject Code :** BCS306A

Question Number	Solution	Marks Allocated
2 c)	Datatype [ ][ ] arrayVariablename = new Datatype[size][size]; Explanation.	2
3 a)	Garbage Collection mechanism	3
	Java heap division – Young, Tenured, Perm area	3
b)	void area(int l, int b) //Rectangle { } void area (int r) //Circle { } void area (int side) //Triangle { } 3 overloaded methods with respective logic	2*3 = 6
	Calling methods in main	2
	Reading inputs	2
c)	General form / Syntax of a class	2
	Example	2
	class classname { access specifier datatype instance variables/ class variables; access specifier returntype methodname(args) { } }	
	OR	
4 a)	i. this keyword: to call constructor within another constructor; to return an object; to resolve namespace collision	3
	ii. Static keyword : Used to create class variable. It can be prefixed to variables, methods and blocks	3
b)	Creating a class 'Employee'	2
	Declaring instance variables	2
	Defining read() and write() methods	2
	Reading five employee information + displaying	4
c)	Types of Constructors — default	2
	— parameterized	2

Subject Title : Object Oriented Programming with Java Subject Code : BCS306A

Question Number	Solution	Marks Allocated
5 a)	Super Keyword Uses : To make a ~~class~~ Call to superclass Constructor To refer to superclass member & resolve namespace collision	Explanation - 4 Example - 2
	Dynamic method dispatch	Explanation - 4

b)	interface Resizable
	{
	public void resize(int ~~radius~~);
	}
	class Circle implements Resizable
	{ int radius;
	void resize(int radius)
	{
	this.radius = radius;
	System.out.println("New Radius" + radius);
	}
	}
	Main Class & main method
		Interface - 3
		Class - 4
		- 3

	OR	
6 a)	Method Overloading & Method Overriding Static polymorphism Dynamic polymorphism	Differences - 6 Example - 2
b)	Inheritance definition - One of the feature of Object Oriented programming that acquires the ~~fo~~ properties of one class into another (super to sub class) Types of Inheritance — Single, Multiple (Not available in Java), Multilevel, Hybrid, Hierarchical	- 1 - 3
c)	Creating Shape class Creating 3 sub classes with methods Demonstrating in Main method	- 1 - 6 - 1

Subject Title : Object Oriented Programming with Java Subject Code : BCS306A

Question Number	Solution	Marks Allocated
7 a)	Explanation of access specifies with visibility control with inheritance & without inheritance in packeage	10
b)	class LowBalance extends Exception { double amt; public LowBalance(double amt) { this.amt = amt; } String toString() { return "Hold on! Low Balance"; } } class Account { : void withdraw(double amt) throws LowBalance { if(bal-amt < minbal) { LowBalance e = new LowBalance(bal-minbal); throw e; } else { bal = bal - amt; }} : }	4 (custom exception) 4 Main method - 2
	OR	1
8 a)	Exception definition Exception handling keywords: try, catch, throw, throws & finally with syntax	9
b)	creating package package balance; public class Account { public double bal; public void displayBalance() { System.out.println(bal); } }	5

	Importing package
	import balance.*;
	public class Main {
	public static void main(String args[])
	{ Account ac = new Account();
	ac.displayBalance();
	}}
		5

Subject Title : Object Oriented Programming with Java Subject Code : BCS306A

Question Number	Solution	Marks Allocated
9 a)	Thread is a smallest unit of dispatchable code that shares the same address space and exists within a process. Ways of creating a thread: 1. By implementing Runnable interface 2. By extending Thread class	1 Listing - 1 4 (2*2)
b)	Synchronization is achieved in 2 ways: 1. Using Synchronized methods 2. Using Synchronized blocks	3*2 (6)

c)	class Main {
	public static void main (String args[])
	{ Integer i = new Integer(100);
	int a = i;
	System.out.println(a);
	}}
		8

	OR	
10 a)	Explanation of Type Wrappers with example Type wrappers converts primitive into object and object into primitive.	6

b)	Integer i = new Integer(100);
	if(i < 100) { // Unboxing internally
	System.out.println(i); }
	Also expression →
		Explanation + Example 6

c)	class MyThread extends Thread
	{ MyThread(String name)
	{ super(name); start(); }
	public void run()
	{ // Business Logic }
	}
	public class Main
	{ Public Static void main(String args[]) {
	MyThread t1 = new MyThread("Thread1");
	MyThread t2 = new MyThread("Thread2");
	}}
		creating MyThread class } 3
		Main class - 2
		creating thread objects & demonstrating concurrency } 3