16 AUG 2024

22MCA204

USN | 1 | B | Y | | | | | | | | |

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(An Autonomous Institute affiliated to Visvesvaraya Technological University, Belagavi)

SEMESTER END EXAMINATION QUESTION PAPER

Second Semester MCA Degree Examination

Regular / Make-up / Arrears / Supplementary

JAVA PROGRAMMING

Time: 3 hrs. **Max. Marks:** 100

*Note: 1. Answer **FIVE** full questions, choosing **ONE** full question from each module.*

Q. No	Module – 1	Marks	CO, RBT
1a.	How Java Programming overcomes the limitation of 'C' Programming? Justify your answer.	10	CO1, K2
b.	Develop a Java Class illustrates the concept of data abstraction and encapsulation. Trace the same.	10	CO1, K2
	OR		
2a.	Exemplify the object oriented programming principles of JAVA.	10	CO1, K2
b.	How to call super class constructors and super class members using super? Demonstrate with an example.	10	CO1, K2
	Module – 2		
3a.	How an exception handling is more useful in building error free Java applications? Justify your views with a suitable Java code.	10	CO2, K2
b.	Differentiate between throw versus throws and final versus finally with a suitable example.	10	CO2, K2
	OR		
4a.	Demonstrate the implementing concepts of packages and interfaces in Java.	10	CO2, K2
b.	Develop a Java class for small finance application and handle suitable exception handling.	10	CO2, K2
	Module – 3		
5a.	Exemplify how inter-thread communication can be achieved in multithreading using producer and consumer problem	10	CO3, K2
b.	With neat diagram, discuss in detail of the life cycle of Java Thread.	10	CO3, K3
	OR		

6a.	Construct a Java program to create multiple threads in Java by implementing runnable interface.	10	CO3, K2
b.	Define Java enumeration? Write a Java program to create an enumeration Day of Week with seven values SUNDAY through SATURDAY, Add a method isworkday() to the DayofWeek class that returns true if the value of which it is called is MONDAY through FRIDAY, otherwise false.	10	CO3, K3
	Module – 4		
7a.	List and explain the methods involved in the servlet life cycle.	10	CO4, K2
b.	Apply the Java Servlet to demonstrate a servlet program to check whether the session is new or old.	10	CO4, K2
	OR		
8a.	List all the attributes of JSP page directive tag and explain any four with an example for each.	10	CO4, K2
b.	Write a JSP program to generate sum of even numbers between 1 and 20.	10	CO4, K2
	Module – 5		
9a.	In detail, discuss the JDBC life cycle with a snippet code.	10	CO5, K2
b.	Develop a Java code to do validation of user credentials against the stored values in the database.	10	CO5, K2
	OR		
10a.	With a neat diagram, explain the states of life cycle of a stateless session Bean.	10	CO5, K2
b.	Write short notes on i. Entity Java Bean ii. Session Bean iii. Message Driven Bean	10	CO5, K2

Course Outcomes (COs):

COs	At the end of the course, the student will be able to
CO-1	Demonstrate the basic programming constructs of Java and OOP concepts to develop Java applications.
CO-2	Illustrate the concepts of generalization and run time polymorphism to develop reusable components.
CO-3	Exemplify the usage of Multithreading in building efficient applications.
CO-4	Build web applications using Servlets and JSP.
CO-5	Design applications using JDBC and Enterprise Java Beans.
K1- Remembering K2 - Understanding K3 – Applying K4- Analyzing K5 - Evaluating K6 -Creating	

"Success is the progressive realization of a worthy goal."

****

DESCRIPTIVE QUESTIONS PATTERN

COURSE CODE: 22MCA204 COURSE NAME: Java Programming

Module-I

Q. No	Scheme and Solutions	Marks
Q1a)	Listing C - programming features — 05M. Justification how Java overcome limitation of C programming - 05M **C supports** - Procedural language - compiler dependent - Supports structs, union, pointers etc. **Java supports** - OOPs features - Garbage collections - String handling - Multithreading etc.	10M.
Q1b)	Coding of each - 02 x 5M **Data abstraction** abstract class shape { String color; abstract double area(); Public Shape(String color) { this.color = color; } class circle extends Shape { double radius; public Circle(String clr, double r) { super(clr); this.radius = r; } } **Encapsulation** class Encap { private String name; Public String getName() { return name; } Public void setName(String newName) { name = newName; } public class Demo { public static void main() { Encap obj = new Encap(); obj.setName("BMSIT"); System.out.println("Name is " + obj.getName); } }	10M

1/11

DESCRIPTIVE QUESTIONS PATTERN

COURSE CODE: 22MCA204 COURSE NAME ... Java Programming .....

Q. No	Scheme and Solutions	Marks
Q2a)	Listing of OOP features – 05M. Each explanation & example – 05M – Data Abstraction & ex:- abstract class A{--} – Encapsulation & ex:- class{--} – Inheritance & ex – class A{} class B extends A{--} – Polymorphism & ex. void A(){} – overloading overriding	10M.
Q2b)	Explanation – 05M Example – 05M Super class constructor can be called using super() from the subclass constructor, and must be the first statement. class A { public A(){ S.O.P("constructor called");} class B extends A { public B(){ super(); S.O.P("subclass constructor called"); public static void main(String args[]) { B b = new B(); } }	10M

2/11

DESCRIPTIVE QUESTIONS PATTERN

COURSE CODE: 22MCA20 COURSE NAME ..... Java Programming .....

Module - 2

Q. No	Scheme and Solutions	Marks

Q3a)	Explanation of how exception handling helpful to build error free applications in Java — 05M
	Justification — 03M
	Code — 02M
	Snippet code
	`try { int z = int x / int y }`
	`} catch ( Exception e ) { . }`
		10M

Q3b)	Differences of each — 02 x 5M throw versus Throws	10M

– throw is used within the method – unchecked exceptions can be handled	throws is used with the method signature – both checked & unchecked exceptions can be handled

final versus finally

– final is used to declare a variable, method, or class as unchangeable.	finally is used after a try or catch block to execute code that must be run regardless of whether an exception is thrown or not.

(08)

3/11

DESCRIPTIVE QUESTIONS PATTERN

COURSE CODE: 22MCA204 COURSE NAME ... Java Programming

Q. No	Scheme and Solutions	Marks

4 a)	Demonstration — 05M
	Coding — 05M
	A java package is a group of similar types of classes, Interfaces & sub-packages.
	Code :- // Package
	package P1;
	class Demo { public void m1() {
	SOP("Method 1"); }}
	Interface is similar to a class, but it contains only abstract methods
	Code Interface Example {
	public void m1();
	public void m2();
	<type> var = value;
	}
		10M

4b)	Coding of Small finance in Java — 05M
	Exception handling — 05M.
	Snippet code
	withdraw = balance - amount;
	// mathematically above statement is correct, but in real time, it's required to check the withdrawal & balance maintainance. So exception handling used for this.
	try { w = b-a; } catch (Exception e) { }
		10M

4/11

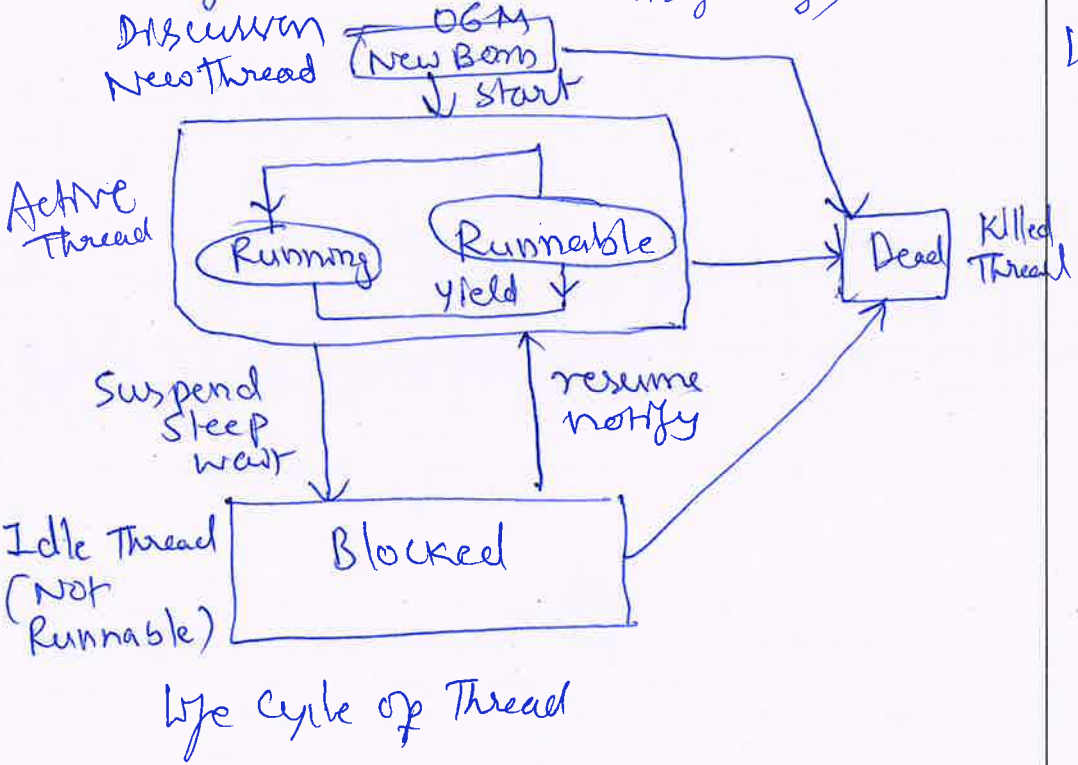
DESCRIPTIVE QUESTIONS PATTERN

COURSE CODE: 22MCA204 COURSE NAME ...Java Programming.....

Module -3

Q. No	Scheme and Solutions	Marks

Q5a)	Inter-Thread communication snippet code — 10M
	
	class Producer extends Thread
	{
	private Shop s;
	Public Producer(Shop c)
	{ s=c; }
	Public void run()
	{ for(int i=0; i<10; i++)
	{ s.put(i); }
	}
	}
	
	class Consumer extends Thread
	{
	private Shop s;
	Public Consumer(Shop c)
	{ s=c; }
	Public void run()
	{
	int val = 0;
	for(int i=0; i<10; i++)
	{ s.get(); }
	}
	}
		10M

Q5b)	Diagram — 04M (Sample given, it may vary) Discussion — 06M  Life Cycle of Thread	10M

5/11

DESCRIPTIVE QUESTIONS PATTERN

COURSE CODE: 22MCA204 COURSE NAME ..... Java Programming .....

Q. No	Scheme and Solutions	Marks

Q6a)	Java program to create multiple threads by implementing runnable interface –
	Correct Syntax – 02M
	Code – 08M
	
	Class MultiThread implements Runnable {
	Public void run()
	{ try
	{ System.out.println("Thread " + Thread.currentThread().getId() + " is running");
	} catch (Exception e) {
	System.out.println("Exception is caught");
	} } }
	
	class MainThread {
	Public static void main (String[] args)
	{ int n = 8; // number of Threads
	for (int i=0; i<n; i++) {
	Thread obj = new Thread (new MultiThread());
	obj.start();
	} } }
		10M

Q6b)	Definition of enumeration – 02M enumeration in java is a special data type which contains a set of predefined constants. ex:- enum Direction { East, West, North, South }	

6/4

DESCRIPTIVE QUESTIONS PATTERN

COURSE CODE: 22MCA204 COURSE NAME Java Programming

Q. No	Scheme and Solutions	Marks

Q6(b)	Continued ——— — 08M
	Java code for enumeration.
	class EnumDemo {
	Public enum Dayofweek { Sunday(1), Monday(2), Tuesday(3), Wednesday(4), Thursday(5), Friday(6), Saturday(7) };
	Public int val;
	EnumDemo(int val)
	{ this.val = val;
	}
	boolean IsWeekDay()
	{ if (val < 6) return true;
	else return false;
	}
	Public static void main (String [] args)
	{ EnumDemo day;
	System.out.println(EnumDemo.Sunday.
	IsWorkDay());
	System.out.println(EnumDemo.Thursday.
	IsWeekDay());
	}
	}
		10M

7/11

DESCRIPTIVE QUESTIONS PATTERN

COURSE CODE: 22MCA204 COURSE NAME ..Java Programming.....

Module-4

Q. No	Scheme and Solutions	Marks

Q7a)	Writing - 02M
	Servlet lifecycle explanation - 08M
	i. Load servlet class
	ii. Create servlet instance
	iii. Call the init(--) method
	iv. Call the service method (-,-)
	v. Call the destroy() method
	import javax.servlet.http.*;
	public class DemoServlet extends HttpServlet
	{ public void doGet (HttpServletRequest req, HttpServletResponse res)
	throws ServletException, IOException {
	res.setContentType("text/html");
	PrintWriter pw = res.getWriter();
	pw.println("<html><body>----");
	pw.close();
	}
	}
		10M

Q7b)	Syntax - 02M
	Full Java code - 08M
	Snippet code.
	HttpSession session = req.getSession();
	session.setAttribute("name", n);
	public long getLastAccessedTime() { }
	public String getId() { }
	public HttpSession getSession(boolean b);
	{ }
		10M

8/11

DESCRIPTIVE QUESTIONS PATTERN

COURSE CODE: 22MCA204 COURSE NAME ..... Java Programming .....

Q. No	Scheme and Solutions	Marks
8a)	Listing all attributes – 02M Example – 04M explanation – 04M – Page directive <%@ page attribute="value" %> → import, contentType, extends, info, buffer, language, isELIgnored, isThreadSafe, autoFlush, session, pageEncoding, errorPage, isErrorpage. ex: <%@ page import="java.util.Date" %> Today is <%= new Date() %> <%@ page contentType="application/msword" %> – – – – .	10M

8b)	JSP Program syntax – 02M
	coding – 08M
	<html><head><title> JSP Prg </title>
	</head><body>
	  displaying sum
	<% int sum = 0;
	for (int num = 1; num <= 20; num++)
	{
	~~sum = sum + num;~~
	if (num % 2 == 0)
	{ sum = sum + num;
	}
	out.println("sum is" + sum);
	}
	%> </body></html>
		10

9/11

DESCRIPTIVE QUESTIONS PATTERN

COURSE CODE: 22MCA204 COURSE NAME: Java Programming

Module-5

Q. No	Scheme and Solutions	Marks
9a)	JDBC Life cycle – 04M Explanation – 06M Begin → Import java.sql package → Load driver → create connection → create statement → execute Statement → clare connection → End.	10M

9b)	Snippet code – 08M
	Syntax – 02M
	Connection con = "dbconnection";
	String un = req.getParameter("username");
	String pwd = req.getParameter("password");
	Statement st = "select un, pwd from login";
	Resultset rs = con.execute(st);
	if (rs.next)
	{ out.println("user found");
	}
	out.println("Mismatch");
		10M

	(OR)	
10a)	Diagram – 02M Explanation – 08M 	10M

10/11

DESCRIPTIVE QUESTIONS PATTERN

COURSE CODE: 22MCA204 COURSE NAME ..... Java Programming .....

Q. No	Scheme and Solutions	Marks
10b)	Each short notes i - Entity Java Bean (EJB) - 04M ii - Session Bean - 03M iii - Message Driven Bean (MDB) - 03M * EJB is a remote object that manages persistent data, Performs complex business logic, potentially uses several dependent Java objects. * Session Bean represent a process or flow. - Each session Bean is associated with one EJB client at time. - Session Bean have stateless & stateful. * MDB is to exist within a pool and to receive & process incoming messages from a message provider. - - . — X End X —	10M

11/11