

Java Coding Answers

Group A Answer All the Questions (5 x 1 = 5)			
1	How many primitive data types are there in JAVA?	R	CO1
2	Which access specifier can be used for a class so that its members can be accessed by a different class in the same package?	R	CO2
3	What is the correct ordering for importing a class?	R	CO3
4	Name some of the most common types of exceptions that might occur in java.	R	CO5
5	What is join() method?	R	CO5
Group B Answer All the Questions (5 x 2 = 10)			
6	What are the features of Java? Explain in brief.	R	CO1
7	What is the difference between <i>while</i> and <i>do-while</i> loop?	R	CO2
8	Define constructor with a suitable example.	R	CO3
9	What is multithreading?	R	CO5
10	Why doesn't Java support Multiple Inheritance?	R	CO4
Group C Answer All the Questions (7 x 5 = 35)			
11	What is typecasting? Why it is required in the program?	R	CO1
12	Develop a Java program in java to print the following triangle:	AP	CO2

	1 01 101 0101 10101		
13	Briefly explain Static binding and Dynamic binding	U	CO3
14	Explain ArrayList and Vector with examples.	U	CO4
15	Explain the life cycle of JAVA thread.	U	CO5
16	Develop a java program to use the yield(), stop() and sleep() methods of a thread.	AP	CO5
17	Develop an applet that receives three numeric values as input from the user and then displays the largest of the three on the screen.	AP	CO6

Group A Answer All 5 Questions (5 x 2 = 10)			
		Knowledge Level	Course Outcome
✓	Define constructor with a suitable example.	Remembering	CO1
✓	Write the syntax to take the input from the user using BufferedReader class.	Remembering	CO1
✓	List the restrictions of using the “super” keyword.	Remembering	CO3
✓	Define multiple inheritance with a suitable block diagram.	Remembering	CO3
✓	What is a finally block? When and how is it used?	Remembering	CO2
Group B Answer All 5 Questions (5 x 4 = 20)			
✓	i) What do you mean by exception handling mechanism? ii) Develop a java program to implement the concept of method overloading. [1 + 3]	Remembering Applying	CO2
✓	i) What is method overriding? ii) Develop a java program to restrict method overriding. [1 + 3]	Remembering Applying	CO1
✓	i) What is hierarchical inheritance? ii) Develop a java program to implement hierarchical inheritance using “super” and “this” keyword. [1 + 3]	Remembering Applying	CO2
✓	i) Define interface with a suitable example. ii) Develop a java program to define your own static methods and call it from main () method. [1 + 3]	Remembering Applying	CO3
10(a)	i) Define an exception called “NoMatchException” that is thrown when a string is not equal to “India”. ii) Develop a java program that uses this exception. [1 + 3]	Remembering Applying	CO2

OR														
10(b)	i) Define nesting of methods. ii) Develop a java program to implement the concept of nesting of methods.	[1 + 3]	Remembering Applying	CO2										
Group C Answer All 2 Questions (2 x 10 = 20)														
11(a)	i) Develop a java program to set and retrieve the priority in a thread. ii) Design an applet to display the bar charts with the data shown in the below table.		Applying Creating	CO4 CO1										
<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>Year</th><th>2016</th><th>2017</th><th>2018</th><th>2019</th></tr> <tr> <td>Total Income (Rs. Crores)</td><td>150</td><td>120</td><td>180</td><td>200</td></tr> </table> <p style="text-align: right;">[5 + 5]</p>					Year	2016	2017	2018	2019	Total Income (Rs. Crores)	150	120	180	200
Year	2016	2017	2018	2019										
Total Income (Rs. Crores)	150	120	180	200										
OR														
11(b)	i) Develop an applet that receives three numeric values as input from the user and then displays the sum and average of the three on the screen. Design a HTML page and test the applet. ii) Develop a java program to throw your own exception.		Applying	CO5										
12(a)	i) Develop a java program to use yield (), stop () and sleep () methods of a thread. ii) Explain different stages in the life cycle of a thread with a suitable block diagram.	[5 + 5]	Applying Understanding	CO5 CO4										
OR														
12(b)	i) Develop a java applet program that draws a Pie Chart of number of students registered in various departments at 2 nd year level within the University of Melbourne. Please access input that are passed through PARAM statements in HTML. Hint: Each pie can be drawn using the drawArc () or the fillArc () method. ii) Develop a simple calculator applet that can handle basic math functions.	[5 + 5]	Applying	CO5										

Programming Questions-

JAVA Questions

- Develop a Java Program to print the following triangle-

1

01

101

0101

10101

```
public class RightTrianglePattern {  
    public static void main(String a[]) {  
        int n = 5;  
  
        for (int i = 1; i <= n; i++) {  
            for (int j = 1; j <= i; j++) {  
                if ((i + j) % 2 == 0) {  
                    System.out.print("1");  
                } else {  
                    System.out.print("0");  
                }  
            }  
            System.out.println();  
        }  
    }  
}
```

- Develop a Java Program to use the yield(), stop() and sleep() methods of a thread

```
public class ThreadMethods extends Thread {  
  
    public void run() {  
        System.out.println("Thread started: " + Thread.curren  
        Thread.yield();  
        System.out.println("Thread after yield(): " + Thread.  
  
        try {  
            Thread.sleep(2000);  
            System.out.println("Thread after sleep(): " + Thre  
        } catch (InterruptedException e) {  
            System.out.println("Thread interrupted while slee  
        }  
}
```

```

        Thread.currentThread().stop();
        System.out.println("Thread after stop(): " + Thread.c
    }

    public static void main(String[] args) {
        ThreadMethods thread = new ThreadMethods();
        thread.start();
    }
}

```

- Develop a Applet Program that receives three numeric values as input from the user and then displays the largest of three on the screen

- Develop a Java Program to implement the concept of method overloading

```

class MethodOverloading {
    public void print(int a) {
        System.out.println("Integer: " + a);
    }

    public void print(double a) {
        System.out.println("Double: " + a);
    }

    public void print(String a) {
        System.out.println("String: " + a);
    }

    public static void main(String[] args) {
        MethodOverloading obj = new MethodOverloading();
        obj.print(10);
        obj.print(10.2);
        obj.print("Hello");
    }
}

```

- Develop a Java Program to restrict method overriding

```

class ParentClass {
    public final void display() {
        System.out.println("This is a final method in the Parent Class");
    }

    public void show() {
        System.out.println("This is a non-final method in the Parent Class");
    }
}

class ChildClass extends ParentClass {
    @Override
    public void show() {
        System.out.println("This is an overridden method in the Child Class");
    }
}

public class RestrictMethodOverriding {
    public static void main(String[] args) {
        ChildClass child = new ChildClass();
        child.display();
        child.show();
    }
}

```

- Develop a Java Program to implement hierarchical inheritance using 'this' and 'super' keyword

```

class Animal {
    String type;

    Animal(String type) {
        this.type = type;
    }

    void displayType() {
        System.out.println("Type of animal: " + type);
    }
}

```

```

}

class Dog extends Animal {
    String breed;

    Dog(String type, String breed) {
        super(type);
        this.breed = breed;
    }

    void displayBreed() {
        System.out.println("Breed of dog: " + breed);
    }
}

public class HierarchicalInheritance {
    public static void main(String[] args) {
        Dog dog = new Dog("Mammal", "Labrador");
        dog.displayType();
        dog.displayBreed();
    }
}

```

- Develop a Java Program to define your own static methods and call it from `main()` method

```

public class MyOwnStaticMethod {
    public static int square(int num) {
        return num * num;
    }

    public static int sum(int a, int b) {
        return a + b;
    }

    public static void main(String[] args) {
        int result1 = square(5);
        System.out.println("Square of 5: " + result1);
    }
}

```

```

        int result2 = sum(10, 20);
        System.out.println("Sum of 10 and 20: " + result2);
    }
}

```

- Define an exception called 'NoMatchException' when a string is not equal to 'India'. Develop a Java Program that uses this exception.

```

class NoMatchException extends Exception {
    public NoMatchException(String message) {
        super(message);
    }
}

public class NoMatchExceptionExample {

    public static void checkString(String input) throws NoMatchException {
        if (!input.equals("India")) {
            throw new NoMatchException("String does not match 'India'");
        } else {
            System.out.println("String matches 'India'");
        }
    }

    public static void main(String[] args) {
        try {
            checkString("USA");
        } catch (NoMatchException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }

        try {
            checkString("India");
        } catch (NoMatchException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}

```

```
    }  
}
```

- Develop a Java Program to implement Nesting of Methods

```
public class NestingofMethods {  
  
    public void outerMethod() {  
        System.out.println("Inside outerMethod");  
        innerMethod();  
    }  
  
    private void innerMethod() {  
        System.out.println("Inside innerMethod");  
        innermostMethod();  
    }  
  
    private void innermostMethod() {  
        System.out.println("Inside innermostMethod");  
    }  
  
    public static void main(String[] args) {  
        NestingofMethods obj = new NestingofMethods();  
        obj.outerMethod();  
    }  
}
```

- Develop a Java Program to set and retrieve the priority in a thread

```
class MyThread extends Thread {  
    public void run() {  
        System.out.println("Thread running: " + getName());  
        System.out.println("Priority of " + getName() + " is  
    }  
}  
  
public class ThreadPriority {  
    public static void main(String[] args) {
```

```

        MyThread t1 = new MyThread();
        MyThread t2 = new MyThread();
        MyThread t3 = new MyThread();

        t1.setName("Thread 1");
        t2.setName("Thread 2");
        t3.setName("Thread 3");

        t1.setPriority(Thread.MIN_PRIORITY);
        t2.setPriority(Thread.NORM_PRIORITY);
        t3.setPriority(Thread.MAX_PRIORITY);

        t1.start();
        t2.start();
        t3.start();
    }
}

```

- Develop a Applet Program to show the bar charts of the given data-

Year	2016	2017	2018	2019
Total Income (in Cr)	150	120	180	200

- Develop a Applet Program that receives the three numeric values from the user and displays the sum and average of all three numbers on screen. Design both HTML and the test applet

- Develop a Java Program to throw your own exception

```

class CustomException extends Exception {
    public CustomException(String message) {
        super(message);
    }
}

```

```

public class MyOwnException {
    public static void validateAge(int age) throws CustomException {
        if (age < 18) {
            throw new CustomException("Age must be at least 18");
        } else {
            System.out.println("Valid age: " + age);
        }
    }

    public static void main(String[] args) {
        try {
            validateAge(15);
        } catch (CustomException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}

```

- Develop a Applet Program that draws a pie chart of number of students registered in various departments at 2nd yr level in a university. Please access Input that are passed through PARAM statements in HTML. (**Hint: Each pie can be draw with drawArc() or fillArc() method**)

- Develop a Applet Program to make a calculator that can handle basic math functions

- Develop a Java Program to make method overriding compulsory

```

abstract class Shape {
    public abstract void draw();
}

class Circle extends Shape {
    public void draw() {
        System.out.println("Drawing a Circle");
    }
}

```

```

}

class Square extends Shape {
    public void draw() {
        System.out.println("Drawing a Square");
    }
}

public class MethodOverridingCompulsory {
    public static void main(String[] args) {
        Shape shape1 = new Circle();
        Shape shape2 = new Square();

        shape1.draw();
        shape2.draw();
    }
}

```

- Develop a Java Program to implement multiple inheritance.

```

interface A {
    void execute(int num1);
}

interface B {
    void execute(int num1);
}

class C implements A, B {
    public void execute(int num1) {
        System.out.println("Hello.. From Implementation Class");
    }
}

public class MultipleInheritance {
    public static void main(String[] args) {
        C obj = new C();
        obj.execute(16);
    }
}

```

```
    }  
}
```

Group A Answer All 5 Questions (5 x 1 = 5)		Knowledge Level (BL1-6)	Course Outcome (CO1-5)
✓	What is a final keyword?	L1	CO4
✓	How can we declare a 2-D array in java having unequal column size?	L1	CO1
✓	Explain Hierarchical inheritance.	L2	CO3
✓	List the restrictions of using the "super" keyword.	L1	CO3
✓	Describe constructor with suitable example.	L4	CO1
Group B Answer Any 5 Questions (5 x 3 = 15)			
✓	Develop a java program to implement the concept of method overloading.	L3	CO2
✓	Create a java program to implement hierarchical inheritance using "super" and "this" keyword.	L6	CO5
✓	Develop a java program to define your own static methods and call it from main () method.	L3	CO4
✓	Develop a java program to implement the concept of nesting of methods.	L3	CO1
10	Develop a java program to make method overriding compulsory.	L3	CO2
11	Design a java program to implement the concept of multiple inheritance.	L6	CO3

1

Must Read Topics for OOP:

Constructor, Constructor Overloading, this, super, final, finally, abstract keywords related to variables, methods and block, Method Overloading, Method Overriding, Restrict Method Overriding, Making Method Overriding Compulsory, Static Methods, User Defined Static Methods, Nesting of Methods,

Inheritance, Implementation of all 5 types of Inheritance, Exception Handling,
Different types of Exception Handling Program,
Threads, Creating Threads by extending Thread Class and by Implementing
Runnable Interface, Thread Life Cycle, Program using stop(), sleep (), yield ()
methods, Thread Priority, Assigning and Retrieving Thread Priority, Applet,
Applet Life Cycle, Design of Applet to display bar charts and taking values as
inputs to find and display mathematical operations.