**Technical Paper**

1. **Which of the following code blocks will cause a compilation error?**public class Test {  
    static {  
    System.out.println("Static Block");  
    }  
    {  
    System.out.println("Instance Block");  
    }  
    Test() {  
    System.out.println("Constructor");  
    }  
    public static void main(String[] args) {  
    Test t = new Test();  
    }  
   }

**Output :**

1. **Which of the following access modifiers allows visibility only within the same package?**

A. public C. default  
B. protected D. private

1. **What is the output of the following code?**public class Main{

public static void main(String[] args){

int x = 5;

int y = ++x \* 10 + x--;

System.out.println(y);

}

}

**Output :**

1. **What will be the result of this code?**class A {  
    void display() {  
    System.out.println("A");  
    }  
   }  
   class B extends A {  
    void display() {  
    System.out.println("B");  
    }  
    void print() {  
    super.display();  
    }  
   }  
   public class Main {  
    public static void main(String[] args) {  
    new B().print();  
    }  
   }

**Output :**

1. **Which of the following is true about static blocks in Java?**

A. Static blocks execute before constructors  
B. Static blocks can access non-static members  
C. Multiple static blocks are not allowed  
D. Static blocks are executed every time an object is created

1. **Which keyword is used to prevent method overriding?**

A. private  
B. static  
C. final  
D. abstract

1. **What does the 'this' keyword refer to in Java?**

A. The current class name  
B. The current package  
C. The current object reference  
D. A static context

1. **Choose the correct statement about constructor overloading:**

A. It allows multiple constructors with the same parameter types  
B. Overloaded constructors must have different names  
C. Constructors can be overloaded by changing number/type of parameters  
D. Constructor overloading is not allowed in Java

1. **What is the output of this?**final class A {

void show() {

System.out.println("This is class A");

}

}

class B extends A {

void display() {

System.out.println("This is class B");

}

}

public class Test {

public static void main(String[] args) {

B b = new B();

b.show();

b.display();

}

} **Output :**

1. **What is the output of this?**class Test {  
    int a = 10;  
    Test() {  
    a = a++ + ++a;

}  
 public static void main(String[] args) {  
 Test t = new Test();  
 System.out.println(t.a);  
 }}

**Output :**

1. **Choose the correct output:**class Test {  
    static int count = 0;  
    Test() {  
    count++;  
    }  
    public static void main(String[] args) {  
    Test t1 = new Test();  
    Test t2 = new Test();  
    System.out.println(Test.count);  
    }  
   }

**Output :**

1. **What is true about method overloading in Java?**

A. Return type must be different  
B. Method name must be different  
C. Parameters must differ  
D. Access modifier must be different

1. **Which concept allows treating subclass objects as superclass objects?**

A. Encapsulation  
B. Polymorphism  
C. Inheritance  
D. Abstraction

1. **Which of the following statements about abstract classes is false?**

A. Abstract classes cannot be instantiated  
B. Abstract class can have constructors  
C. Abstract class can be declared final  
D. Abstract class can have both abstract and non-abstract methods

1. **What is the output of the following code?**class Parent {  
    static void show() {  
    System.out.println("Parent");  
    }  
   }  
   class Child extends Parent {  
    static void show() {  
    System.out.println("Child");  
    }  
   }  
   public class Test {  
    public static void main(String[] args) {  
    Parent p = new Child();  
    p.show();  
    }  
   } **Output :**
2. **Which of the following is not a pillar of OOP?**

A. Encapsulation  
B. Inheritance  
C. Compilation  
D. Polymorphism

1. **What will be the output?**class A {  
    A() {  
    System.out.println("A Constructor");  
    }  
   }  
   class B extends A {  
    B() {  
    System.out.println("B Constructor");  
    }  
   }  
   class C extends B {  
    C() {  
    System.out.println("C Constructor");  
    }  
   }  
   public class Test {  
    public static void main(String[] args) {  
    new C();  
    }  
   }

**Output :**

1. **Which of the following statements about constructors is false?**

A. A constructor can be private  
B. A constructor must have the same name as the class  
C. A constructor can return a value  
D. A class can have multiple constructors

1. **Which is the correct way to achieve abstraction in Java?**

A. Using final keyword  
B. Using abstract classes and interfaces  
C. Using static methods only  
D. Hiding main method

1. **What is the output?**public class Test {  
    public static void main(String[] args) {  
    System.out.println(10 + 20 + "Java" + 10 + 20);  
    }  
   }

**Output :**

**CODING**

1. **Write a Java program to count number of even and odd elements from given array?**

**Input: 3 4 7 9 2 6 1  
Output:  
 Even elements: 3  
 Odd elements: 4**

1. **Write a Java program to display prime elements from given array?**

**Input: int arr[] = {5, 9, 3, 11, 15, 7, 14};  
Output: 5 3 11 7**

1. **Write a Java program to display array elements in descending order without using sort method?**

**Input: 5 9 3 11 15 7 14  
Output: 15 14 11 9 7 5 3**

1. **Write a Java program to display duplicate and unique elements from given array?**

**Input: 3 6 1 2 3 9 4 4 7 6 10  
Output: Uniques: 1 2 9 7 10**

**Duplicates: 3 6 4**

1. **Write a Java program to display lucky number from given array?**

**Input: 1 2 2 3 3 3  
Output: 3**

1. **Write the difference between for loop and for-each loop with a program?**
2. **Define Encapsulation and write a program for it?**
3. **Define Polymorphism and its types. Write a program for method overloading and overriding?**
4. **Write a Java program to achieve multiple inheritance using interface?**
5. **Write a program for static block, instance block and constructor, and explain how the output came?**