Core Java Training
Chapter 3: Java Programming

What is the incorrect statement about JDK, JRE, JVM and JIT
JIT is Just In Time Compiler and it is inside the JVM
Java Compiler is included in JDK (Java Development Kit)
JRE contains JVM and the java class libraries and the run-time libraries
JVM contains JRE and the java class libraries and the run-time libraries

What is true about JDK and JRE?
When JDK is installed JRE is included along with it
When JRE is installed JDK is included along with it
JRE contains the java compiler for compiling the java source code
Developers only need JRE for writing the programs, JDK is needed for the users of the program

What is JDK and what is JRE?
JDK is Java Development Knowledge and JRE is Java Runtime Executor
JDK is Java Development Knowledge and JRE is Java Runtime Environment
JDK is Java Development Kit and JRE is Java Runtime Environment
None of the above

What will be the output of the following program.

```
class helloworld
(
    public static void main(String[] u)
    {
        System.out.println("Payilagam");
    }
)

public class Test {
    int a = 10;
   public void doStuff(int a) {
        a += 1;
        System.out.println(a++);
    }
    public static void main(String args[]) {
        Test t = new Test();
        t.doStuff(3);
    }
}
```

What will be the output of the following program.

```
class HelloWorld
{
    public static void main(String st[])
    {
        System.out.println("Hello Payilagam");
    }
}
```

Chapter 4: Data Types, Variables and Operators

Guess the output:

```
int Output = 10;
boolean b1 = false;
if((b1 == true) && ((Output += 10) == 20))
{
    System.out.println("We are equal " + Output);
}
else
{
    System.out.println("Not equal! " + Output);
}
```

In the following pieces of code, A and D will compile without any error. True or false?

```
A: StringBuffer sb1 = "abcd";
B: Boolean b = new Boolean("abcd");
C: byte b = 255;
D: int x = 0x1234;
E: float fl = 1.2;

class Value
{
    public int i = 15;
}
public class Test
{
    public static void main(String argv[])
    {
        Test t = new Test();
        t.first();
    }
    public void first()
    {
        int i = 5;
        Value v = new Value();
        v.i = 25;
        second(v, i);
        System.out.println(v.i);
    }
    public void second(Value v, int i)
    {
        i = 0;
        v.i = 20;
```

```
        Value val = new Value();
        v =  val;
        System.out.println(v.i + " " + i);
    }
}
```

What results from attempting to compile and run the following code?

```
public class Ternary
{
   public static void main(String args[])
   {
      int a = 5;
      System.out.println("Value is - " + ((a < 5) ? 9.9 : 9));
   }
}
```

What is the result of compiling and running the following code?

```
public class Tester {
        static void test(float x) {
                System.out.print("float");
        }
        static void test(double x) {
                System.out.print("double");
        }
        public static void main(String[] args) {
                test(99.9);
        }
}

public class Tester {
        static void test(float x) {
                System.out.print("float");
        }
        static void test(double x) {
                System.out.print("double");
        }
        public static void main(String[] args) {
                test((float) 99.9);
        }
}

public class Tester {
        static void test(float x) {
                System.out.print("float");
        }
        public static void main(String[] args) {
                test(99.9);
        }
}

class Main {
   public static void main(String args[]) {
        int t;
        System.out.println(t);
   }
}

class Test {
   public static void main(String[] args) {
     for(int i = 0; 0; i++)
     {
        System.out.println("Hello");
        break;
     }
   }
}

class Test
{
   public static void main(String[] args)
   {
      Double object = new Double("2.4");
      int a = object.intValue();
      byte b = object.byteValue();
      float d = object.floatValue();
      double c = object.doubleValue();
       System.out.println(a + b + c + d );
    }
}

public class NumberSystem{
   public static void main(String[] args){

      int hexVal = 0x1a;
      System.out.println("Value : " + hexVal);


   }
}

public class NumberSystem{
   public static void main(String[] args){
      int val = 0b11010;
      System.out.println("Value : " + val);
   }
}
```

After the below declaration, What is the value of c[50]?

```
char[] c = new char[100];
```

What will be the result of calling the following method with an input of 2?

```
 public int adder( int N ){
 return 0x100 + N++ ;
 }
```

What happens when you attempt to compile and run the following code?

```
 public class Logic {
 static int minusOne = -1 ;
 static public void main(String args[] ){
   int N = minusOne >> 31 ;
   System.out.println("N = " + N );
 }
}
```

What is the output of the below?

```
public int MaskOff( int n ){
   return n | 3 ;
 }
```

How many String objects are created in the following code?

```
String A, B, C ;
 A = new String( "1234" ) ;
 B = A ;
 C = A + B ;
```

Which of the following versions of initializing a char variable would cause a compiler error? [Check all correct answers.]

```
char c = - 1 ;
char c = '\u00FF' ;
char c = (char) 4096 ;
char c = 4096L ;
char c = 'c' ;
char c = "c" ;
```

What happens when you try to compile and run the following code?

```
 public class EqualsTest{
  public static void main(String args[]){
   Long LA = new Long( 7 ) ;
   Long LB = new Long( 7 ) ;
   if( LA == LB )
   System.out.println("Equal");
   else System.out.println("Not Equal");
  }
 }

public class EqualsTest{
  public static void main(String args[]){
   char A = '\u0005' ;
   if( A == 0x0005L ) {
     System.out.println("Equal");
   }
   else {
    System.out.println("Not Equal");
   }
  }}
```

What would happen if you tried to compile and run the following code?

```
public class EqualsTest{
  public static void main(String args[]){
   Long L = new Long( 7 );
   if( L.equals( 7L ))
   System.out.println("Equal");
   else System.out.println("Not Equal");
  }
 }

public class EqualsTest{
  public static void main(String args[]){
   Object A = new Long( 7 );
   Long L = new Long( 7 ) ;
   if( A.equals( L ))
   System.out.println("Equal");
   else System.out.println("Not Equal");
  }
 }

class Test{
  public static void main(String args[]){
    try{
      byte x = 7;
      byte y = 5;
      System.out.println(x | y);
    }catch(Exception e){
      System.out.println(
                "Exception Thrown");
    }//end catch
  }//end main()
}//end class definition

class Test{
```

```
  public static void main(
                    String args[]){
    try{
      boolean x = true;
      boolean y = false;
      System.out.print((x & y) + " " +
                       (x & x));
      System.out.print("  ");
      System.out.print(
        (x ^ y) + " " + (x ^ x) + " " +
                       (y ^ y));
      System.out.print("  ");
      System.out.println(
        (x | y) + " " + (x | x) + " " +
                       (y | y));

    }catch(Exception e){
      System.out.println(
                "Exception Thrown");
    }//end catch
  }//end main()
}//end class definition
```

class Test{

```
  public static void main(String args[]){
    try{
      byte x = 1;
      boolean y = false;
      System.out.println((boolean)x & y);
    }catch(Exception e){
      System.out.println("Exception Thrown");
    }//end catch
  }//end main()
}//end class definition
```

class Test{

```
  public static void main(String args[]){
    String x = new String("100");
    String y = new String("100");
    if(x == y)
      System.out.println("Equal");
    else
      System.out.println("Not Equal");
  }//end main()
}//end class definition
```

class Test{

```
  public static void main(String args[]){
    String x = new String("100");
    String y = new String("100");
    if(x.equals(y))       System.out.println("Equal");
    else       System.out.println("Not Equal");
  }//end main()
}//end class definition
```

class Test{

```
  public static void main(String args[]){
    String x = "100";
    String y = "100";
    if(x == y)       System.out.println("Equal");
    else       System.out.println("Not Equal");
  }//end main()
}//end class definition
```

What value is assigned to x as the result of executing the following code?

```
int y = 3;
y++;
x = y++;
```

What will be the output of the following program?

```
public class ShortTest {
   public static void main(String[] args) {
      boolean x = true;
      boolean y = false;
      if (x || y) { System.out.println(true);
      } else { System.out.println(false);
      }
   }
}
```

class OperatorsOutput

```
{
   public static void main(String s[])
   {
      int a = 12 + 21 * 3 - 9 / 2;
      int b = 14 - 32 * 4 + 175 / 8 - 3;
      if(++a > 71 && --b < 20)
      {
         System.out.println("a = " + a + " b = " + b);
      }
      if(b-- == -97 || a-- < 100)
      {
         System.out.println("a = " + a + " b = " + b);
      }
   }
}
```

```
public class Doubt {
   public static void main(String s[]) {
      int x = 20;
      int y = 25;
      if (++x < (y = y -= 4) || (x = x += 4) > y) {
         System.out.println(x + "," + y);
      }
   }
}
```

```
public class DemoOnFloat
{
   public static void main(String[] args)
   {
      float fl = 5.3f;
      if (fl == 5.3)
         System.out.println("Both are equal");
      else
         System.out.println("Both are not equal");
   }
}
```

```
class Animals
{
   public static void main(String [] args)
   {
      boolean rabbit = true;
      boolean donkey = false;
      boolean leporidae = true;
      if (rabbit & donkey | donkey & leporidae | donkey)
         System.out.print("DOG ");
      if (rabbit & donkey | donkey & leporidae | donkey | rabbit)
         System.out.println("CAT ");
   }
}
```

```
class PrintRelation
{
   public static void main(String s[])
   {
      int a = 7 * 3 + 6 / 2 - 5;
      int b = 21 - 8 + a % 3 * 11;
      if(a < b)
      {
         System.out.println("A is less than B");
      }
      if(a = b)
      {
         System.out.println("A is equal to B");
      }
      if(a > b)
      {
         System.out.println("A is greater than B");
      }
   }
}
```

```
class MyClass1
{
   public static void main(String s[])
   {
      boolean a, b, c;
      a = b = c = true;
      if( !a || ( b && c ) )
      {          System.out.println("If executed");       }
      Else       {          System.out.println("else executed");
      }
   }
}
```

```
public class Blue {
   public static void main(String s[]) {
      int x = 20;
      int y = 25;
      if (++x < (y = y -= 4) || (x = x += 4) > y) {
         System.out.println(x + " " + y);
      }
   }
}
```

```
class StudentPass
{
   public static void main(String s[])
   {
      int marks = 80;
      if( marks > 70 )
         System.out.println("Distinction");
      if( marks > 35 )
         System.out.println("Pass");
      else
         System.out.println("Fail");
         System.out.println("Better luck next time");
   }
}
```

```
class FindWinner2
{
   public static void main(String s[])
```

```
    {
        int india_score = 300;
        int pakistan_score = 290;

        System.out.println( india_score > pakistan_score ? "India Wins" :
"Pakistan Wins");
    }
}
```

What will be the output of the following program? Assume that the argument passed as - Payilagam.

```
public class MyName {
    public static void main(String args[]) {
        if (args.length == 0 || args[0].equalsIgnoreCase("Payilagam")) {
            System.out.println("I Dont Know");
        } else {
            System.out.println(args[0]);
        }
    }
}
```

What will be the output of the following program?

```
class Test1
{
    public static void main(String s[])
    {
        float f = 75.0f;
        double d = 75.0;
        int i = 75;
        if( f == d )
        {
            if( f == i )
            {
                System.out.println("f, d and i are equal");
            }
            else
            {
                System.out.println("f, d are equal but i is not equal");
            }
        }
        else
        {
            System.out.println("f and d are not equal");
        }
    }
}
```

class DetermineGroup

```
{
    public static void main(String s[])
    {
        boolean male = false;
        int age = 30;
       if( male )
            if( age < 20 )
                System.out.println("Boy");
            else
                System.out.println("Man");
        else
            if( age < 20 )
                System.out.println("Girl");
            else
                System.out.println("Woman");
    }
}
```

class DetermineGroup

```
{
    public static void main(String s[])
    {
        boolean male = false;
        int age = 30;
        if( male )
            if( age < 20 )
                System.out.println("Boy");
            else
                System.out.println("Man");
        else
            if( age < 20 )
                System.out.println("Girl");
            else
                System.out.println("Woman");
    }
}
```

class StudentPass

```
{
    public static void main(String s[])
    {
        int marks = 80;
        if( marks > 70 )
            System.out.println("Distinction");
            System.out.println("Congratulations");
        else if( marks > 35 )
            System.out.println("Pass");
        else
            System.out.println("Fail");
            System.out.println("Better luck next time");
```

```
    }
}

public void output(boolean a, boolean b) {
    if (a && b) {
        System.out.println("A && B");
    } else if (a) {
        System.out.println("A");
    } else {
        if (!b) {
            System.out.println("notB");
        } else {
            System.out.println("ELSE");
        }
    }
}
```

class WeekDays

```
{
    public static void main(String s[])
    {
        int day = 7;
        switch(day)
        {
            case 1:
                System.out.println("Monday");
            case 2:
                System.out.println("Tuesday");
            case 3:
                System.out.println("Wednesday");
            case 4:
                System.out.println("Thursday");
            case 5:
                System.out.println("Friday");
        }
    }
}
```

class NumberDescription

```
{
    public static void main(String s[])
    {
        int number = 5;
        switch(number)
        {
            case 2:
            case 4:
            case 6:
            case 8:
                System.out.println("Even");
                break;
            case 1:
            case 3:
            case 5:
            case 7:
            case 9:
                System.out.println("Odd");
                break;
            case 2:
            case 3:
            case 5:
            case 7:
                System.out.println("Prime");
                break;
            case 4:
            case 9:
                System.out.println("Perfect Square");
                break;
            default:
                System.out.println("Can only describe numbers from 1 to 9");
        }
    }
}
```

class HappyNewYear

```
{
    public static void main(String s[])
    {
        int code = 3;
         switch(code)
        {
            case 1:
                System.out.println("Wish");
            case 2:
                System.out.println("You");
            default:
                System.out.println("A");
            case 3:
                System.out.println("Happy");
            case 4:
                System.out.println("New");
            case 5:
                System.out.println("Year");
        }
    }
}
```

class Directions

```
{
   public static void main(String s[])
   {
      char direction = 'N';
      char west = 'W';
      switch(direction)
      {
         case 'N':
            System.out.println("North");
            break;
         case 'E':
            System.out.println("East");
            break;
         case west:
            System.out.println("West");
            break;
         case 'S':
            System.out.println("South");
      }
   }
}
```

```
public class Test {
   public static void main(String args[]) {
      int i = 1, j = 0;
      switch (i) {
         case 2 :
            j += 6;
         case 4 :
            j += 1;
         default :
            j += 2;
         case 0 :
            j += 4;
      }
      System.out.println("j = " + j);
   }
}
```

```
public class KIs {
   public static void main(String args[]) {
      int k = 66;
      switch (k) {
         default :
            System.out.print("Website");
         case 66 :
            System.out.print("Payilagam");
         case 'k' :
            System.out.print("Chennai");
         case 'j' :
            System.out.print("Java");
            break;
      }
   }
}
```

```
public class Prize {
   public static void main(String args[]) {
      int x = 111, y = 101;
      switch (x & y) {
         case 1 :
            System.out.println("-1-");
         case 101 :
            System.out.println("-101-");
         case 111 :
            System.out.println("-111-");
         case 010 :
            System.out.println("-010-");
         default :
            System.out.println("-" + x & y + "-");
      }
   }
}
```

```
public class DemoOnSwitch
{
   public static void main(String[] args)
   {
      int var = 12;
      switch (var)
      {
         case 014 :
            System.out.print("Hello");
            break;
         case 12 :
            System.out.print("Hi");
         default :
            System.out.print("How are you?");
      }
   }
}
```

```
public class TestSwitch
{
   public static void main(String a[])
   {
      char ch = 'a';
      switch (ch)
```

```
    {
       case 'a' :
       case 'A' :
          System.out.print(ch);
          break;
       case 'b' :
       case 'B' :
          System.out.print(ch);
          break;
       case 'c' :
       case 'C' :
          System.out.print(ch);
          break;
       case 'd' :
       case 'D' :
          System.out.print(ch);
    }
  }
}

  class SelectionStatements {
    public static void main(String args[])
    {
       int var1 = 5;
       int var2 = 6;
       if ((var2 = 1) == var1)
          System.out.print(var2);
       else
          System.out.print(++var2);
    }
  }

  class CommaLearningDemo {
    public static void main(String args[])
    {
        int sum = 0;
        for (int i = 0, j = 0; i < 5 & j < 5; ++i, j = i + 1)
           sum += i;
             System.out.println(sum);
    }
  }

  class JumpStatments {
    public static void main(String args[])
    {
        int x = 2;
        int y = 0;
        for ( ; y < 10; ++y) {
            if (y % x == 0)
               continue;
            else if (y == 8)
                break;
            else
              System.out.print(y + " ");
        }
    }
  }

class Output {
      public static void main(String args[])
      {
        final int a=10,b=20;
       while(a<b)
       {
        System.out.println("Hello");
       }
       System.out.println("World");
      }
  }

   class Output {
      public static void main(String args[])
      {
          int a = 5;
          int b = 10;
          first: {
            second: {
              third: {
                 if (a ==  b >> 1)
                    break second;
                  }
              System.out.println(a);
            }
            System.out.println(b);
          }
      }
  }

public class Example {
  public static void main(String[] args) {
    byte b = 3;
    b = -b;
    System.out.println(b);
  }
}

public class Example {
```

```
  public static void main(String[] args) {
    double d = 259;
    byte b = (byte)d;
    System.out.println(b);
  }
}

public class Example {
  public static void main(String[] args) {
    String s = "456";
    int i = 123;
    System.out.println((int)i + s);
  }
}

public class Example {
  public static void main(String[] args) {
    String s = "123";
    Object o = s;
    s = (String)o;
    System.out.println(s);
  }
}

public class Conversions
{
   public static void main(String args[])
   {
      byte x;
      int a = 270;
      double b = 128.128;
      x = (byte) a;
      System.out.println("a and x " + a + " " + x);
      a = (int) b;
      System.out.println("b and a " + b + " " + a);
      x = b;
      System.out.println("b and x " + b + " " + x);
   }
}

class IntegerConversion
{
   public static void main(String args[])
   {
      long l = 55;
      int i = 44;
      short s = 33;
      byte b    = 22;
      i = (int) l;
      s = (short) i;
      b = (byte) s;
      System.out.println("l = " + l);
      System.out.println("i = " + i);
      System.out.println("s = " + s);
      System.out.println("b = " + b);
   }
}

public class IAmGreat {
   public static void main(String args[]) {
      int i = 132;
      short s = 15;
      byte b = (byte) i;
      int x = b + s;
      System.out.println("Value of x is " + x);
   }
}

public class DemoOnbyte
{
   public static void main(String[] args)
   {
      byte a = -0x15;
      byte c = (a >> 4);
      System.out.print(c);
   }
}

public class DemoOnbyte
{
   public static void main(String[] args)
   {
      byte a = -0x15;
      byte c = (a >> 4);
      System.out.print(c);
   }
}


class IntegerGroupAddition
{
   public static void main(String args[])
   {
      long l = 30;
      int i = 50;
      short s = 60;
      byte b = 70;
```

```
        int sum = l + i + s + b;
        System.out.println("Sum = " + sum);
    }
}
```

class SumOfDoubles

```
{
    public static void main(String args[])
    {
        double d1 = 0.7;
        double d2 = 77.0/1000.0;
        double d3 = d1 * 10;
        double d4 = 77E+2;
        double d5 = 77E-5;
        double sum = d1 + d2 + d3 + d4 + d5;
         System.out.println("sum = " + sum);
    }
}
```

Chapter 5: Classes and Objects

Predict the output of following Java program?

```
class Test {
  int i;
}
class Main {
   public static void main(String args[]) {
     Test t;
     System.out.println(t.i);
}

class demo
{
    int a, b;
    demo()
    {
        a = 10;
        b = 20;
    }
    public void print()
    {
        System.out.println ("a = " + a + " b = " + b + "\n");
    }
}


class Test
{
    public static void main(String[] args)
    {
        demo obj1 = new demo();
        demo obj2 = obj1;
        obj1.a += 1;
        obj1.b += 1;
        System.out.println ("values of obj1 : ");
        obj1.print();
        System.out.println ("values of obj2 : ");
        obj2.print();
    }
}
```

Predict the output of following Java program.

```
class demoClass
{
    int a = 1;
     void func()
    {
        demo obj = new demo();
        obj.display();
    }


    class demo
    {
        int b = 2;
        void display()
        {
            System.out.println("\na = " + a);
        }
    }


    void get()
    {
        System.out.println("\nb = " + b);
    }
}
```

class Test

```
{
    public static void main(String[] args)
    {
        demoClass obj = new demoClass();
        obj.func();
        obj.get();
    }
}

class Test
```

```
{
   int a = 1;
   int b = 2;
   Test func(Test obj)
   {
      Test obj3 = new Test();
      obj3 = obj;
      obj3.a = obj.a++ + ++obj.b;
      obj.b = obj.b;
      return obj3;
   }

   public static void main(String[] args)
   {
      Test obj1 = new Test();
      Test obj2 = obj1.func(obj1);
      System.out.println("obj1.a = " + obj1.a + "  obj1.b = " + obj1.b);
      System.out.println("obj2.a = " + obj2.a + "  obj1.b = " + obj2.b);
   }
}

class Test
{
   int a = 1;
   int b = 2;
   Test func(Test obj)
   {
      Test obj3 = new Test();
      obj3 = obj;
      obj3.a = obj.a++ + ++obj.b;
      obj.b = obj.b;
      return obj3;
   }
   public static void main(String[] args)
   {
      Test obj1 = new Test();
      Test obj2 = obj1.func(obj1);
      System.out.println("obj1.a = " + obj1.a + "  obj1.b = " + obj1.b);
      System.out.println("obj2.a = " + obj2.a + "  obj1.b = " + obj2.b);
   }
}

class A
{
   static int i;
   static
   {
      System.out.println(1);
      i = 100;
   }
}

public class StaticInitializationBlock
{
   static
   {
      System.out.println(2);
   }
   public static void main(String[] args)
   {
      System.out.println(3);
      System.out.println(A.i);
   }
}

public class A
{
   static
   {
      System.out.println(1);
   }

   static
   {
      System.out.println(2);
   }

   static
   {
      System.out.println(3);
   }

   public static void main(String[] args)
   {
      A a;
   }
}

class ClassOne
{
   {
      System.out.println(1);
   }

   static
   {
```

```
      System.out.println(2);
   }

   public ClassOne(int i)
   {
      System.out.println(3);
   }

   public ClassOne()
   {
      System.out.println(4);
   }
}

public class ClassTwo
{
   {
      System.out.println(5);
   }

   public static void main(String[] args)
   {
      System.out.println(6);
      ClassOne one = new ClassOne();
      ClassOne two = new ClassOne(10);
   }
}

class ClassOne
{
   char c = 'A';
   {
      c = 'B';
   }
   public ClassOne(char c)
   {
      this.c = c;
   }
}
```

What will be the output of the following program?

```
class ReferencesAndObjects
{
   public static void main(String s[])
   {
      Student st1 = new Student();
      Student st2 = new Student();
      st1 = st2;
      st1.name = "Rajesh";
      st2.marks = 87;
      st1.section = 'C';
      System.out.println("Print using st1 : " + st1.name + " " + st1.marks +
" " + st1.section);
      System.out.println("Print using st2 : " + st2.name + " " + st2.marks +
" " + st2.section);
   }
}
class Student
{
   String name;
   int marks;
   char section;
}
```

What will be the output of the following program?

```
class ReferencesAndObjects
{
   public static void main(String s[])
   {
      Student st1 = new Student();
      Student st2 = new Student();
      st2 = st1;
      st2 = new Student();
      st1.name = "Rajesh";
      st2.marks = 87;
      st1.section = 'C';
      System.out.println("Print using st1 : " + st1.name + " " + st1.marks +
" " + st1.section);
      System.out.println("Print using st2 : " + st2.name + " " + st2.marks +
" " + st2.section);
   }
}

class Student
{
   String name;
   int marks;
   char section;
}
```

What will be the output of the following program?

```
class ReferencesAndObjects
{
   public static void main(String s[])
   {
      Student st1 = new Student();
      Student st2 = new Student();
```

```
        st2 = st2;
        st2 = new Student();
        st1.name = "Rajesh";
        st2.marks = 87;
        st1.section = 'C';
        System.out.println("Print using st1 : " + st1.name + " " + st1.marks +
" " + st1.section);
        System.out.println("Print using st2 : " + st2.name + " " + st2.marks +
" " + st2.section);
    }
}

class Student
{
    String name;
    int marks;
    char section;
}


public class ClassTwo
{
    public static void main(String[] args)
    {
        ClassOne one = new ClassOne('Z');
        System.out.println(one.c);
    }
}

class T {
  int t = 20;
}
class Main {
   public static void main(String args[]) {
      T t1 = new T();
      System.out.println(t1.t);
   }
}

class T {
  int t = 20;
  T() {
    t = 40;
  }
}
class Main {
   public static void main(String args[]) {
      T t1 = new T();
      System.out.println(t1.t);
   }
}

class Test
{
    static int a;

    static
    {
        a = 4;
        System.out.println ("inside static block\n");
        System.out.println ("a = " + a);
    }

    Test()
    {
        System.out.println ("\ninside constructor\n");
        a = 10;
    }

    public static void func()
    {
        a = a + 1;
        System.out.println ("a = " + a);
    }

    public static void main(String[] args)
    {
        Test obj = new Test();
        obj.func();
    }
}

public class ClassB {
    int x = 3;
    public ClassB() {
        System.out.print(new ClassA().method());
    }
    public ClassB(int i) {
        System.out.print(i);
    }
    public int method(int i){
        return x + i;
    }
}
public class ClassA {
    int y;
    public ClassA() {}
```

```
    public ClassA(ClassB classB) {
        System.out.print(classB.method(1));
    }
    public int method(){
        System.out.print(new ClassB(4).method(3));
        return y;
    }
    public static void main(String[] args){
        ClassA classA = new ClassA(new ClassB(new ClassB().method(2)));
    }
}
```

Chapter 6: Exploring Methods

```
class Main {
    public static void main(String args[]) {
            System.out.println(fun());
    }
    int fun()
    {
      return 20;
    }
}
```

```
    public static void main(String args[]) {
        String x = null;
        giveMeAString(x);
        System.out.println(x);
    }
    static void giveMeAString(String y)
    {
        y = "Payilagam";
    }
}
```

In the following method declaration, what is the return type?public static int myMethod(int count, double value) { return 4;}

In the following method declaration, what is the name of the method? public static void showMenu(String category) {}

In the following method declaration, how many formal parameters are there?public static double squareRoot(double value) { return 2.3;}

In the following method how many values are returned?public static void syncPhone(int idNumber) {}

Write a method declaration (just the first line but include the {) for the following description. The wording must be exact and correct!The method must be called calculateFinalExam. The method must return a double. The method accepts 3 parameters:1. the studentId which must be an integer2. the studentName which must be a String3. the testScore which must be a double.

If we do not provide String array as the argument to the main method as shown below, then the program compiles but throws a run-time error "NoSuchMethodError". Say True or False

```
class First
{
    public static void main()
    {
        System.out.println("Hello Payilagam");
    }
}
```

If static modifier is removed from the signature of the main method as shown below, then the program compiles but throws a run-time error "NoSuchMethodError". Say True or False

```
class Second
{
    public void main(String []args)
    {
        System.out.println("Hello World");
    }
}
```

What will be the output of the following program?

```
public class HaiTwice {
    static int num;
    public static void main(String[] args) {
        HaiTwice p = new HaiTwice();
        p.start();
        System.out.println(num);
    }
    void start() {
        int var = 7;
        twice(var);
        System.out.print(var + " ");
    }
    void twice(int var) {
```

```
        var = var * 2;
        num = var;
    }
}

public class MyObject {
    public static void main(String[] args) {
        MyObject.myStaticMethod();
        System.out.println(MyObject.myStaticMethod());
    }

    public String memberVar;
    static private String memberStaticVar;
    static public String myStaticMethod() {
        memberVar = "Value";
        memberStaticVar = "Value";
        MyObject obj = new MyObject();
        obj.memberVar = "Value";
        System.out.println("Have a nice to Participants");
        return ("No Error");
    }
}

class MethodCalls
{
    public static void main(String[] args)
    {
        a1(8);
    }

    static int a1(int a1)
    {
        System.out.print(" a1 = " + a1);
        return a2(a1++);
    }

    static int a2(int a2)
    {
        if (a2 == 0)
        {
            return 10;
        }
        System.out.print(" a2 = " + a2);
        return a1(a2 / 2);
    }
}

public class PrintStatment10
{
    public static void main(String args[])
    {
        PrintStatment10 obj = new PrintStatment10();
        obj.printBinaryFormat(23);
    }
    public void printBinaryFormat(int number)
    {
        int binary[] = new int[25];
        int index = 0;
        while (number > 0)
        {
            binary[index++] = number % 2;
            number = number / 2;
        }
        for (int i = index - 1; i >= 0; i--)
        {
            System.out.print(binary[i]);
        }
    }
}

class A {
    public static void main(String arg[]) {
        B a = new B();
        B b = new B();
        B c = new B();
        B d = new B();
        System.out.println("i = " + a.i);
        System.out.println("d = " + b.d);
        System.out.println("c = " + c.c);
        System.out.println("s = " + d.s);
    }
}
class B {
    int i;
    double d;
    char c;
    String s;
}

class Test
{
    public static void main(String args[])
    {
        for (int x = 0; x < 4; x++)
        {
            System.out.println(x);
        }
        System.out.println(x);
```

```
   }
}

class Test
{
   public static void main(String args[])
   {
      int x;
      for (x = 0; x < 4; x++)
      {
         System.out.println(x);
      }
     System.out.println(x);
   }
}

class Test
{
   public static void main(String args[])
   {
      int a = 5;
      for (int a = 0; a < 5; a++)
      {
         System.out.println(a);
      }
   }
}

public class LocalPrimitiveVariableInit {
     int age;
   float salary;
   String name;
   public void sayHello(){
      String message = "This is local variable";
      String reason = " beacause the variable is in a method";
      System.out.println("Hello " + message + reason);
   }


   public static void main(String[] args) {
      String anotherMessage = "This is another local variable in main
method";
      int hoursWorked;
    System.out.println("Hours worked : " + hoursWorked);
      LocalPrimitiveVariableInit variableInit = new
LocalPrimitiveVariableInit();
      System.out.println("Default Salary is " + variableInit.salary);
      variableInit.sayHello(); // call the method and print message
      System.out.println(anotherMessage);
   }
}

class San
{
 public void m1 (int i,float f)
 {
  System.out.println(" int float method");
 }


 public void m1(float f,int i);
  {
  System.out.println("float int method");
  }


  public static void main(String[]args)
  {
    San s=new San();
      s.m1(20,20);
  }
}

class overload {
      int x;
           int y;
      void add(int a) {
         x =  a + 1;
      }
      void add(int a, int b){
         x =  a + 2;
      }
   }
   class Overload_methods {
      public static void main(String args[])
      {
         overload obj = new overload();
         int a = 0;
         obj.add(6);
         System.out.println(obj.x);
      }
  }


class overload {
      int x;
           int y;
      void add(int a){
         x =  a + 1;
      }
      void add(int a , int b){
```

```
        x =  a + 2;
      }
   }
   class Overload_methods {
      public static void main(String args[])
      {
         overload obj = new overload();
         int a = 0;
         obj.add(6, 7);
         System.out.println(obj.x);
      }
  }

class overload {
      int x;
           double y;
      void add(int a , int b) {
         x = a + b;
      }
      void add(double c , double d){
         y = c + d;
      }
      overload() {
         this.x = 0;
         this.y = 0;
      }
   }
   class Overload_methods {
      public static void main(String args[])
      {
         overload obj = new overload();
         int a = 2;
         double b = 3.2;
         obj.add(a, a);
         obj.add(b, b);
         System.out.println(obj.x + " " + obj.y);
      }
  }

class test {
      int a;
      int b;
      void meth(int i , int j) {
         i *= 2;
         j /= 2;
      }
   }
   class Output {
      public static void main(String args[])
      {
         test obj = new test();
               int a = 10;
         int b = 20;
         obj.meth(a , b);
         System.out.println(a + " " + b);
      }
   }
```

class MethodOverloading

```
{
   public static void main(String s[])
   {
      print();
      print(8);
      print(20 < 10);
   }
   public static void print()
   {
      System.out.println("Called print with no parameters");
   }
   public static void print(int i)
   {
      System.out.println("Called print with int parameter");
   }
   public static void print(boolean b)
   {
      System.out.println("Called print with boolean parameter");
   }
}
```

If the following methods are declared in a class, which of them fail to compile because of overloading rules?

```
int multiplication(int a, int b)    // 1
float multiplication(float a, int b)    // 2
float multiplication(int a, float b)    // 3
void multiplication(float a)    // 4
int multiplication(int a)    // 5
void multiplication(int a)    // 6
```

class MethodOverloading

```
{
   public static void main(String s[])
   {
      int i = 8;
      print(i);
      print();
      int j = 9;
```

```
        print(j);
    }

    public static void print()
    {
        System.out.println("Called print with no parameters");
    }

    public static void print(int i)
    {
        System.out.println("Called print with int parameter i");
    }

    public static void print(int j)
    {
        System.out.println("Called print with int parameter j");
    }
}

class MethodOverloading
{
    public static void main(String[] args)
    {
        int a = 12;
        double b = 13;
        double c = m(a, b);
        double d = m(c, a);
        double e = m(a, (int) d);
        System.out.println("c = " + c + " d = " + d + " e = " + e);
    }

    public static double m(int x, double y)
    {
        return x + y;
    }

    public static double m(double x, double y)
    {
        return x - y;
    }

    public static double m(int x, int y)
    {
        return x % y;
    }
}

class MethodOverloading
{
    public static void main(String[] args)
    {
        int a = 12;
        double b = 13;
        double c = m(a, b);
        double d = m(c, a);
        double e = m(a, (int) d);
        System.out.println("c = " + c + " d = " + d + " e = " + e);
    }

    public static double m(int x, double y)
    {
        return x + y;
    }

    public static double m(double x, double y)
    {
        return x - y;
    }

    public static double m(int x, int y)
    {
        return x % y;
    }
}
```

Chapter 7: Inheritance

Question 1:

```
class Base {
    public void show() {
      System.out.println("Base::show() called");
    }
}

class Derived extends Base {
    public void show() {
      System.out.println("Derived::show() called");
    }
}

public class Main {
    public static void main(String[] args) {
        Base b = new Derived();;
        b.show();
    }
}

    final public void show() {
```

```
    System.out.println("Base::show() called");
  }
}
```

Question 2:

```
class Base {
  final public void show() {
    System.out.println("Base::show() called");
  }
}

class Derived extends Base {
  public void show() {
    System.out.println("Derived::show() called");
  }
}

class Main {
  public static void main(String[] args) {
     Base b = new Derived();;
     b.show();
  }
}
```

Question 3:

```
class Base {
  public static void show() {
    System.out.println("Base::show() called");
  }
}

class Derived extends Base {
  public static void show() {
    System.out.println("Derived::show() called");
  }
}

class Main {
  public static void main(String[] args) {
     Base b = new Derived();;
     b.show();
  }
}
```

Question 4:

```
class A
{
  int  b = 50;
}
 class B extends A
{
  int b = 20;
}

public class MainClass
{
  public static void main(String[] args)
  {
     A a = new B();
     System.out.println(a.b);
  }
}
```

Question 5:

```
class Base{
        public Base(){
                System.out.print("Base");
        }
}
public class Derived extends Base{
        public Derived(){
                this("Examveda");
                System.out.print("Derived");
        }
        public Derived(String s){
                System.out.print(s);
        }
        public static void main(String[] args){
                new Derived();
        }
}
```

Question 6:

```
abstract class C1{
            public C1(){
                        System.out.print(1);
            }
}
class C2 extends C1{
            public C2(){
                        System.out.print(2);
            }
}
class C3 extends C2{
            public C3(){
                        System.out.println(3);
            }
```

```
}
public class Test{
        public static void main(String[] a){
                new C3();
        }
}


public class Test
{
   public static void main(String[] args)
   {
    int i = 100;
    long l = i;
    float f = l;
    System.out.println("Int value "+i);
    System.out.println("Long value "+l);
    System.out.println("Float value "+f);
   }
}

public class Test
{
   public static void main(String[] args)
   {
    double d = 100.04;
    long l = (long)d;
    int i = (int)l;
    System.out.println("Double value "+d);
    System.out.println("Long value "+l);
    System.out.println("Int value "+i);
}
}

class IntToByteConversion
{
   public static void main(String arg[])
   {
      int a = 350;
      byte b;
      b = (byte) a;
      System.out.println("b = " + b );
   }
}

class DatatypeCasting
{
   public static void main(String arg[])
   {
      byte b;
      int i = 81;
      double d = 323.142;
      float f = 72.38f;
      char c = 'A';
      c = (char) i;
      System.out.println("i = " + i + " c = " + c);
      i = (int) d; // LINE A
      System.out.println("d = " + d + " i = " + i); // LINE B
      i = (int) f; // LINE C
      System.out.println("f = " + f + " i = " + i); // LINE D
      b = (byte) d;
      System.out.println("d = " + d + " b = " + b);
   }
}
Verify the Output
class Vehicle
{
   int maxSpeed = 120;
}

/* sub class Car extending vehicle */
class Car extends Vehicle
{
   int maxSpeed = 180;
   void display()
   {
      /* print maxSpeed of base class (vehicle) */
      System.out.println("Maximum Speed: " + super.maxSpeed);
   }
}

/* Driver program to test */
class Test
{
   public static void main(String[] args)
   {
      Car small = new Car();
      small.display();
   }
}

What is the output.
/* Base class Person */
class Person
{
   void message()
   {
```

```
        System.out.println("This is person class");
    }
}

/* Subclass Student */
class Student extends Person
{
    void message()
    {
        System.out.println("This is student class");
    }
    // Note that display() is only in Student class
    void display()
    {
        // will invoke or call current class message() method
        message();
        // will invoke or call parent class message() method
        super.message();
    }
}

/* Driver program to test */
class Test
{
    public static void main(String args[])
    {
        Student s = new Student();
        // calling display() of Student
        s.display();
    }
}
```

Check the output

```
/* superclass Person */
class Person
{
    Person()
    {
        System.out.println("Person class Constructor");
    }
}

/* subclass Student extending the Person class */
class Student extends Person
{
    Student()
    {
        // invoke or call parent class constructor
        super();
        System.out.println("Student class Constructor");
    }
}

/* Driver program to test*/
class Test
{
    public static void main(String[] args)
    {
        Student s = new Student();
    }
}
```

Verify the output for the below classes:

```
public Class SuperDemo{
int a,b;
}

public Class Subdemo extends SuperDemo{
int a,b;
void disply(){

System.out.println(a);
System.out.println(b);
super.a=10;
super.b=20;
System.out.println(super.a);
System.out.println(super.b);
}

public static void main (String args[]) {
 Subdemo obj= new Subdemo();
obj.a=1;
obj.b=2;
obj.disply();
}
}
```

Verify the output for the below coding:

```
public Class SuperDemo{
int a,b;
public void show() {
System.out.println(a);
System.out.println(b);
}
}
```

```
public Class Subdemo extends SuperDemo{
int a,b;
void disply(){
System.out.println(a);
System.out.println(b);
super.a=10;
super.b=20;
 super.show();
}

public static void main (String args[]) {
 Subdemo obj= new Subdemo();
obj.a=1;
obj.b=2;
obj.disply();
}
}
```

Verify the output:

```
public Class SuperDemo{
int a,b;
SuperDemo(int x, int y){
 a=x;
b=y
System.out.println("Super class constructor called ");
}
}

public Class Subdemo extends SuperDemo{
int a,b;
SubDemo(int x, int y){
super(10,20);
 a=x;
b=y
System.out.println("Sub class constructor called ");
}

public static void main (String args[]) {
 Subdemo obj= new Subdemo(1,2);
}
}
```

Verify the output:

```
public class ThisDemo {
    int a, b;

 ThisDemo(int a, int b){
      a=a;
      b=b;
 }
   public static void main(String[] args){
       ThisDemo obj= new ThisDemo(1,2);
       System.out.println(obj.a);
       System.out.println(obj.b);
}

public class ThisDemo {
    int a, b;
 ThisDemo(int a, int b){
      this.a=a;
       this.b=b;
 }
   public static void main(String[] args){
       ThisDemo obj= new ThisDemo(1,2);
       System.out.println(obj.a);
       System.out.println(obj.b);
}
}

public class ThisDemo {
    int a, b;

ThisDemo(){
System.out.println("Default constructor called");
}

 ThisDemo(int a, int b){
       this();
        this.a=a;
       this.b=b;
 }

   public static void main(String[] args){
       ThisDemo obj= new ThisDemo(1,2);
       System.out.println(obj.a);
       System.out.println(obj.b);
}
}

public class Test{
    int a, b;
Test(int a, int b){
       this.a=a;
       this.b=b;
 }
```

```
void show(){
System.out.println("Show() method called");
}

void print(){
    this.show();
    System.out.println(a);
    System.out.println(b);
 }
    public static void main(String[] args){
        Test obj= new Test(1,2);
        obj.print()
}
}
```

Guess the output:

```
public class InheritanceTest
{
  public static void main(String[] args)
  {
    Parent c = new Child();
    c.doSomething();
  }
}

class Parent
{
  public void doSomething()
  {
    System.err.println("Parent called");
  }
}

class Child extends Parent
{
  public void doSomething()
  {
    System.err.println("Child called");
  }
}
```

What is the output?

```
public class InheritanceTest
{
  public static void main(String[] args)
  {
    Parent c = new Child();
  }
}

class Parent
{
  public Parent()
  {
    System.err.println("Parent called");
  }
}

class Child extends Parent
{
  public Child()
  {
    System.err.println("Child called");
  }
}
```

Chapter 8: Interfaces and Abstract Classes

Predict the output of the following program.

```
abstract class demo
{
    public int a;
    demo()
    {
        a = 10;
    }
    abstract public void set();
    abstract final public void get();
}

class Test extends demo
{
    public void set(int a)
    {
        this.a = a;
    }
    final public void get()
    {
        System.out.println("a = " + a);
    }

    public static void main(String[] args)
    {
        Test obj = new Test();
        obj.set(20);
        obj.get();
    }
```

```
}
```

What is the output of this program?

```
interface calculate {
    void cal(int item);
}
class display implements calculate {
    int x;
    public void cal(int item) {
        x = item * item;
    }
}
class interfaces {
    public static void main(String args[]) {
        display arr = new display;
        arr.x = 0;
        arr.cal(2);
        System.out.print(arr.x);
    }
}
```

What is the output of this program?

```
interface calculate {
        int VAR = 0;
        void cal(int item);
    }
    class display implements calculate {
        int x;
      public  void cal(int item) {
            if (item<2)
                x = VAR;
            else
                x = item * item;
        }
    }
 class interfaces {
        public static void main(String args[]) {
            display[] arr=new display[3];
           for(int i=0;i<3;i++)
           arr[i]=new display();
           arr[0].cal(0);
           arr[1].cal(1);
           arr[2].cal(2);
           System.out.print(arr[0].x+" " + arr[1].x + " " + arr[2].x);
        }
    }
```

What should be done to compile the code successfully?

```
interface IBeingZeroStudent
{
        double getPercentageMarks();
        String getName();
        int getAge();
}

class BeingZeroStudent implements IBeingZeroStudent
{
        public String getName()
        {
                return "BeingZeroStudent";
        }
        public int getAge()
        {
                return 20;
        }
}
```

What is the output of this program?

```
class A {
    public int i;
    private int j;
}
class B extends A {
    void display() {
        super.j = super.i + 1;
        System.out.println(super.i + " " + super.j);
    }
}
class inheritance {
    public static void main(String args[])
    {
        B obj = new B();
        obj.i=1;
        obj.j=2;
        obj.display();
    }
}
```

What is the output of this program?

```
class A {
    public int i;
    public int j;
    A() {
        i = 1;
        j = 2;
        }
}
```

```
class B extends A {
    int a;
        B() {
        super();
    }
}
class super_use {
    public static void main(String args[])
    {
        B obj = new B();
        System.out.println(obj.i + " " + obj.j)
    }
}
```

What is the output of this program?

```
abstract class A {
    int i;
    abstract void display();
}
class B extends A {
    int j;
    void display() {
        System.out.println(j);
    }
}
class Abstract_demo {
    public static void main(String args[])
    {
        B obj = new B();
        obj.j=2;
        obj.display();
    }
}
```

What is the output of this program?

```
class A {
    int i;
    void display() {
        System.out.println(i);
    }
}
class B extends A {
    int j;
    void display() {
        System.out.println(j);
    }
}
class method_overriding {
    public static void main(String args[])
    {
        B obj = new B();
        obj.i=1;
        obj.j=2;
        obj.display();
    }
}
```

What is the output of this program?

```
class A {
    public int i;
    protected int j;
}
class B extends A {
    int j;
    void display() {
        super.j = 3;
        System.out.println(i + " " + j);
    }
}
class Output {
    public static void main(String args[])
    {
        B obj = new B();
        obj.i=1;
        obj.j=2;
        obj.display();
    }
}
```

What will be the output of the following program?

```
public abstract class AbstractTest
{
    public int getNum()
    {
        return 45;
    }
    public abstract class Bar
    {
        public int getNum()
        {
            return 38;
        }
    }
    public static void main (String [] args)
    {
        AbstractTest t = new AbstractTest()
        {
            public int getNum()
```

```
        {
            return 22;
        }
    };
    AbstractTest.Bar f = t.new Bar()
    {
        public int getNum()
        {
            return 57;
        }
    };
    System.out.println(f.getNum() + " " + t.getNum());
  }
}
```

What will be the output of the following program?

```
public class Test {
  public static void main(String[] args) {
    new Z().method1();
    new Z().method2();
  }
}
abstract class X {
  abstract void method1();
  abstract void method2();
}
abstract class Y extends X {
  void method1() {
    System.out.println("Method1 implemented here.");
  }
}
class Z extends Y {
  void method2() {
    System.out.println("Method2 implemented here.");
  }
}
```

What will be the output of the following program?

```
interface MyFirstInterface {
  void output();
}
interface MySecondInterface extends MyFirstInterface {
  void output();
}
public class DemoOnInterface implements MySecondInterface {
  public void output() {
    System.out.println("Programming competattion on Payilagam...
Hurry up");
  }
  public static void main(String args[]) {
    DemoOnInterface demoOnInterface = new DemoOnInterface();
    demoOnInterface.output();
  }
}
```

What will be the output of the following program?

```
public class CricketPlayersUsingInterfaces {
  public static void main(String s[]) {
    StrongBatsmen sachin = new StrongBatsmen("Sachin", 100, 326);
    sachin.makeCentury();
    sachin.takeWickets();
    StrongBatsmen gambhir = new StrongBatsmen("Gambhir", 25);
    gambhir.makeCentury();
  }
}
class StrongBatsmen implements IBatsmen, IBowler {
  int numberOfCenturies;
  String name;
  int wickets;
  StrongBatsmen(String name, int numberOfCenturies, int wickets) {
    this.numberOfCenturies = numberOfCenturies;
    this.wickets = wickets;
    this.name = name;
  }
  StrongBatsmen(String name, int numberOfCenturies) {
    this.numberOfCenturies = numberOfCenturies;
    this.name = name;
  }
  public void makeCentury() {
    System.out.println(name + " made " + numberOfCenturies + "
centuries.");
  }
  public void takeWickets() {
    System.out.println(name + " taken " + wickets + " wickets.");
  }
}
interface IBatsmen {
  void makeCentury();
}
interface IBowler {
  void takeWickets();
}
```

What will be the output of the following program?

```
class Company
{
  public static void main(String args[])
  {
```

```
        Software.NestedIF nif = new Project();

        if(nif.isNotNegative(10))
            System.out.println("10 is not negative");
        if(nif.isNotNegative(-12))
            System.out.println("This won't be displayed");
    }
}

class Software
{
    public interface NestedIF
    {
        boolean isNotNegative(int x);
    }
}

class Project implements Software.NestedIF
{
    public boolean isNotNegative(int x)
    {
        return x < 0 ? false : true;
    }
}
```

What will be the output of the following program?

```
public class King {
    public static void main(String[] args) {
        King k = new King();
        Elephant e = k.new Elephant();
        System.out.print("Output = ");
        System.out.print(e.step2(2, 3));
    }
    interface Queen {
        float step2(int low, int high);
    }
    interface Pawn {
        float step3(int a, int b, int c);
    }
    abstract class Knight implements Queen, Pawn {
    }
    class Elephant implements Queen {
        public float step2(int x, int y) {
            return 2;
        }
    }
}
```

What will be the output of the following program?

```
interface SixesMachine {
    void hitSixes();
}
abstract class DhoniInTheMaking implements SixesMachine {
    public String numberOfSixes() {
        return "6 0 6 3 6 6 6 6";
    }
    public void printRunsTrail(String runsAndRuns) {
        System.out.println(runsAndRuns);
    }
}
public class Dhoni extends DhoniInTheMaking {
    public static void main(String args[]) {
        DhoniInTheMaking outputClass = new Dhoni();
        outputClass.hitSixes();
    }
    public void hitSixes() {
        numberOfSixes();
        printRunsTrail(numberOfSixes());
    }
}
```

What will be the output of the following program?

```
public class King {
    public static void main(String[] args) {
        King k = new King();
        System.out.print("Output = " + k.new Elephant().step2(2, 3));
    }
    interface Queen {
        float step2(int low, int high);
    }
    interface Pawn {
        float step3(int a, int b, int c);
    }
    abstract class Knight implements Queen, Pawn {
    }
    class Elephant implements Queen {
        public float step2(int x, int y) {
            return (float)(x * 3);
        }
    }
}
```

What will be the output of the following program?

```
public class Revolution2020 {
    public static void main(String[] args) {
        Love love = new Love();
        love.aarti();
        Ambition gopal = new Corruption();
```

```
        Ambition raghav = new Ambition() {
            public String goal() {
                return "Pen Is Powerful ";
            }
            public void aarti() {
                System.out.print("Respect ");
            }
        };
        gopal.aarti(); gopal.goal();
        raghav.aarti(); raghav.goal();
        System.out.print(((Corruption) gopal).shukla());
    }
}
class Love {
    void aarti() { System.out.print("Beautiful ");    }
}
interface Ambition {
    String goal();
    void aarti();
}
class Corruption extends Love implements Ambition {
    int shukla() throws Exception { return 2020; }
    public String goal() { return "Get Rich "; }
    public void aarti() { System.out.print("Confused "); }
}
```

What will be the output of the following program?

```
interface Pet
{
    public void test();
}
class Animal implements Pet
{
    public void test()
    {
        System.out.println("Interface Method Implemented");
        System.out.println("rest of the code");
    }
    public static void main(String args[])
    {
        Pet p = new Animal();
        p.test();
    }
}
```

What will be the output of the following program?

```
public class SmileyTest {
    public static void main(String[] args) {
        Smile a = new Smile();
        talk(a);
        a.frown();
    }
    public static void talk(ISmile ia1) {
        System.out.print(" :-0 ");
        ia1.smile();
    }
}

interface ISmile {
    void smile();
}

class Smile implements ISmile {
    void smile() {
        System.out.print(" :-) ");
    }
    void frown() {
        ISmile a1 = new Smile();
        a1.smile();
        System.out.print(" :-[ ");
    }
}
```

What will be the output of the following program?

```
public class SmileyTest
{
    public static void main(String[] args)
    {
        SmileyTest s = new SmileyTest();
        Smile a = new Smile();
        s.talk(a);
        a.smile();
        ISmile a1 = new Smile();
        ((Smile)a1).frown();
    }

    void talk(Smile ia1)
    {
        ia1.smile();
        System.out.print(" :-0 ");
    }
}

class ISmile
{
    void smile()
    {
        smile();
```

```
    }
    ISmile()
    {
        System.out.print(" :-@ ");
    }
}

class Smile extends ISmile
{
    void smile()
    {
        System.out.print(" :-) ");
    }
    void frown()
    {
        ISmile obj = new Smile();
        System.out.print(" :-[ ");
    }
}


class Small {
    public Small() {
        System.out.print("a ");
    }
}

class Small2  extends Small {
    public Small2() {
        System.out.print("b ");
    }
}

class Small3 extends Small2 {
    public Small3() {
        System.out.print("c ");
    }
}

public class Test {
    public static void main(String args[]) {
        new Small3();
    }
}
```

Given the code. What is the result?

```
class Hotel {
    public int bookings;
    public void book() {
        bookings++;
    }
}

public class SuperHotel extends Hotel {
    public void book() {
        bookings--;
    }


    public void book(int size) {
        book();
        super.book();
        bookings += size;
    }


    public static void main(String args[]) {
        Hotel hotel = new SuperHotel();
        hotel.book(2);
        System.out.print(hotel.bookings);
    }
}
```

Chapter 10: String, StringBuffer, StringBuilder

What is the expected output?

```
public static void main(String[] args) {
 boolean stmt1 = "payilagam" == "payilagam";
 boolean stmt2 = new String("payilagam") == "payilagam";
    boolean stmt3 = new String("payilagam") == new String("payilagam");
 System.out.println(stmt1 && stmt2 || stmt3);
}

public static void main(String[] args) {
 boolean stmt1 = "payilagam" == "payilagam";
    boolean stmt2 = new String("payilagam").equals(new String("payilagam"));
 boolean stmt3 = "payilagam".toString()=="payilagam";
 System.out.println(stmt1 && stmt2 && stmt3);
}
```

Which of the statements would evaluate to true?

```
public class Tester {
```

```
        public static void main(String[] args) {
                StringBuffer sb = new StringBuffer("payilagam");
                String s = new String("payilagam");
                boolean stmt1 = s.equals(sb) ;
                boolean stmt2 = sb.equals(s) ;
                boolean stmt3 = sb.toString() == s ;
                boolean stmt4 = sb.toString().equals(s) ;
                boolean stmt5 = s.equals(sb.toString()) ;
        }
}
```

```
public class Tester {
        public static void main(String[] args) {
                StringBuffer sb1 = new StringBuffer("payilagam");
                StringBuffer sb2 = new StringBuffer("payilagam");
                boolean stmt1 =sb1.equals(sb2) ;
                boolean stmt2 = sb1 == sb2;
                String s1 = new String("payilagam");
                String s2 = new String("payilagam");
                boolean stmt3 = s1.equals(s2);
                boolean stmt4 = s1 == s2;
        }
}
```

```
public static void main(String args []) {
                String stmt = null;
 System.out.print(null+stmt);
 System.out.print(stmt+null);
 }
```

What is the initial capacity of the following string builder?
StringBuilder sb = new StringBuilder("Incredible India!");

Consider the following string:
String word = "How is Chennai Today? It is too hot!";

What is the value displayed by the expression word.length()?
What is the value returned by the method call word.charAt(12)?
Write an expression that refers to the letter C in the string referred to by word.

How long is the string returned by the following expression? What is the string?
"Was it a car or a cat I saw?".substring(9, 12)

In the following program, called ComputeResult, what is the value of result after each numbered line executes?

```
public class ComputeResult {
   public static void main(String[] args) {
      String original = "software";
      StringBuilder result = new StringBuilder("hi");
      int index = original.indexOf('a');

/*1*/   result.setCharAt(0, original.charAt(0));
/*2*/   result.setCharAt(1, original.charAt(original.length()-1));
/*3*/   result.insert(1, original.charAt(4));
/*4*/   result.append(original.substring(1,4));
/*5*/   result.insert(3, (original.substring(index, index+2) + " "));

      System.out.println(result);
   }
}
```

Show two ways to concatenate the following two strings together to get the string "Vanakkam, Chennai!":
String hi = "Vanakkam, ";
String mom = "Chennai!";

Write a program that computes your initials from your full name and displays them.

```
public class Anagram {

   public static boolean areAnagrams(String string1,
                                     String string2) {

      String workingCopy1 = removeJunk(string1);
      String workingCopy2 = removeJunk(string2);

            workingCopy1 = workingCopy1.toLowerCase();
            workingCopy2 = workingCopy2.toLowerCase();

            workingCopy1 = sort(workingCopy1);
            workingCopy2 = sort(workingCopy2);
```

```
    return workingCopy1.equals(workingCopy2);
}

protected static String removeJunk(String string) {
    int i, len = string.length();
    StringBuilder dest = new StringBuilder(len);
                char c;

            for (i = (len - 1); i >= 0; i--) {
                c = string.charAt(i);
                if (Character.isLetter(c)) {
                        dest.append(c);
                }
            }

    return dest.toString();
}

protected static String sort(String string) {
            char[] charArray = string.toCharArray();
            java.util.Arrays.sort(charArray);
    return new String(charArray);
}

public static void main(String[] args) {
    String string1 = "Cosmo and Laine:";
    String string2 = "Maid, clean soon!";

    System.out.println();
    System.out.println("Testing whether the following "
                + "strings are anagrams:");
    System.out.println("    String 1: " + string1);
    System.out.println("    String 2: " + string2);
    System.out.println();

    if (areAnagrams(string1, string2)) {
        System.out.println("They ARE anagrams!");
    } else {
        System.out.println("They are NOT anagrams!");
    }
    System.out.println();
}
```

```
}
```

What will be output of below statements?

```
String s = "Java String Quiz";
System.out.println(s.substring(5,3));
```

What will be output of below statements?

```
String s1 = "Cat";
String s2 = "Cat";
String s3 = new String("Cat");
System.out.println(s1==s2);
System.out.println(s1==s3);
```

What will be the output of below statements?

```
String s1 = null;
System.out.println(s1); //line 2
System.out.println(s1.toString()); //line 3
```

What will be the output of below program?

```
public class Test {
        public static void main(String[] args) {
                String x = "abc";
                String y = "abc";
                x.concat(y);
                System.out.print(x);
        }
}
```

What will be the output of below statements?

```
String s = "Java"+1+2+"Quiz"+""+(3+4);
System.out.println(s);
```

What will be the output of below statements?

```
String s1 = "abc";
String s2 = "def";
System.out.println(s1.compareTo(s2));
```

What will be the output of below statements?

```
String s1 = "abc";
String s2 = new String("abc");
System.out.print(s1==s2);
```

```
System.out.println(s1==s2.intern());
```

What will be the output of below program?

```
public class Test {
	public static void main(String[] args) {
		String s1 = "abc";
		String s2 = "abc";
		System.out.println("s1 == s2 is:" + s1 == s2);
	}
}
```

What will be output of below statements?

```
String s = "Java String Quiz";
System.out.println(s.charAt(s.toUpperCase().length()));
```

What will be the output of below statements?

```
String s1 = "abc";
StringBuffer s2 = new StringBuffer(s1);
System.out.println(s1.equals(s2));
```

What will be the output of below code snippet?

```
String s1 = "abc";
String s2 = new String("abc");
s2.intern();
System.out.println(s1==s2);
```

What will be the output of below code snippet?

```
String s1 = new String("payilagam");
String s2 = new String("PAYILAGAM");
System.out.println(s1 = s2);
```

What is the output?

```
public static void main(String[] args) {
		String s1 = null;
		String s2 = null;
		if (s1 == s2)
			System.out.print("A");
		if (s1.equals(s2))
			System.out.print("B");
	}
```

What is the output?

```
public class Tester {
	public static void main(String[] args) {
		String stmt = "payilagam is here to help you";
		for (String token : stmt.split("//s")) {
			System.out.print(token + " ");
		}
	}
}
```

public class Tester {

```
	public static void main(String[] args) {
		String s = "";
		Integer x = 5;
		StringBuffer sb = new StringBuffer();
		if (x < 15)
			s.concat("payilagam");
		else
			sb.append("payilagam");
		 System.out.print(s + sb);
	}
}
```

public class Tester {

```
	public static void main(String[] args) {
		String s = "";
		Integer x = 5;
		StringBuffer sb = "";
		if (x < 0)
			s.concat("payilagam");
		else
			sb.append("payilagam");
		 System.out.print(s + sb);
	}
}
```

public class Tester {

```
	public static void main(String[] args) {
		Scanner sc = new Scanner("payilagam 2014, true
123");
		while (sc.hasNext()) {
		if (sc.hasNextBoolean())
				System.out.print("Boolean");
			if (sc.hasNextInt())
				System.out.print("Int");
```

```
                    sc.next();
                }
        }
        }

public class Tester {
        public static void main(String[] args) {
                String str = "java";
                StringBuffer sb = new StringBuffer("payilagam");
                sb.insert(9, ".com");
                str.concat("champ");
                if (sb.length() < 6 || str.equals("payilagam")) {
                        System.out.print(sb);
                }
                sb.delete(2, 7);
                System.out.print(sb);
        }
}

public class Test {
        public void method(StringBuffer sb) {
                System.out.println("StringBuffer method");
        }
        public void method(String s) {
                System.out.println("String method");
        }
        public static void main(String[] args) {
                Test test = new Test();
                test.method(null);
        }
}

class Test {
        public static void main(String[] args) {
         String payil = "payilagam_chennai2014chennai";
 Pattern p = Pattern.compile(".{4}c+(m)*"); //line 1
         Matcher m = p.matcher(payil);
         while(m.find()) {
                System.out.print(m.start());
         }
        }
}
```

Determine the output from the following code.

```
String str = "World Wide Web";
for (int i = 0; i < 12; i+=6) {
    System.out.print( str.charAt( i ) );
}
```

What will be the value of str after the following statements are executed.

```
String str = "Dr. Decaffeinated";
StringBuffer strBuf = new StringBuffer(str.substring(6, 10) );
strBuf.setCharAt(1, 'o');
strBuf.append( 'e' );
str = strBuf.toString( );
```

What will be the output of the following program?

```
public class GoodEvening {
   public static void stringReplace(String text) {
      text = text.replace('j', 'c');
   }
   public static void bufferReplace(StringBuffer text) {
      text = text.append("c");
   }
   public static void main(String args[]) {
      String textString = new String("java");
      StringBuffer textBuffer = new StringBuffer("java");
      stringReplace(textString);
      bufferReplace(textBuffer);
      System.out.println(textString + textBuffer);
   }
}
```

What will be the output of the following program?

```
public class StrReplace {
   public static void main(String args[]) {
      StringBuffer sf = new StringBuffer();
      sf.append("I have two pets");
      sf.replace(11, 23, "dogs");
      System.out.print(sf);
   }
}
```

What will be the output of the following program?

```
public class Friends {
```

```
    public static void main(String[] args) {
        String name = "Raju";
        StringBuffer sb = new StringBuffer(name);
        System.out.println(sb);
        sb.replace(2, 4, "vi");
        System.out.print(sb);
    }
}
```

What is the output of the program?

```
public class DailyDoseB {
    public static String overlay(String str, String overlay, int start,
int end) {
        if (str == null) {    return null;    }
        if (overlay == null) {    overlay = "";    }
        int len = str.length();
        if (start < 0) {    start = 0;    }
        if (start > len) {    start = len;    }
        if (end < 0) {    end = 0;    }
        if (end > len) {    end = len;    }
        if (start > end) {
            int temp = start;
            start = end;
            end = temp;
        }
        return new StringBuffer(len + start - end + overlay.length()
+                                                    1).append(str.substring(0,
start)).append(overlay).append(str.substring(end)).toString();
    }
    public static void main(String[] arg) {
        System.out.println(overlay("abcdef", "xxx", 1, 4));
    }
}
```

What will be the output of the following program?

```
public class Campus {
    public static void main(String[] args) {
        String name = "PAYILAGAM";
        StringBuffer sb = new StringBuffer(name);
        sb.substring(3);
        System.out.println(sb);
    }
}
```

What will be the output of the following program?

```
public class Campus {
    public static void main(String[] args) {
        String name = "PAYILAGAM";
        StringBuffer sb = new StringBuffer(name);
        name = sb.substring(3);
        System.out.println(name);
    }
}
```

What will be the output of the following program?

```
public class Sbsubstring {
    public static void main(String args[]) {
        StringBuffer str = new StringBuffer("String control switch");
        String str1 = str.substring(7);
        System.out.println("Buffer: " + str1);
        String str2 = str1.substring(0, 7);
        System.out.print("After: " + str2);
    }
}
```

What will be the output of the following program?

```
public class Console {
    public static void main(String[] arg) {
        int total = 0;
        StringBuffer input = new StringBuffer("MAGALIYAP");
        input.trimToSize();
        String str = input.reverse().toString();
        System.out.println(str);
        System.out.println(input.reverse().toString());
        for (int i = 0; i < str.length(); i++) {
            total += str.codePointAt(i++);
            str = str.substring(i);
        }
        System.out.println(str);
        System.out.print("Total " + total);
    }
}
```

What will be the output of the following program?

```
public class Append {
    public static void main(String args[]) {
```

```
        StringBuffer rb = new StringBuffer();
        rb.append("use pen instead of pencil");
        rb.appendCodePoint(50);
        System.out.print("After append Code: " + rb);
    }
}
```

public class Codepoint {

```
    public static void main(String args[]) {
        StringBuffer sub = new StringBuffer("Mango");
        int c = sub.codePointAt(2);
        System.out.println("Code point at given char: " + c);
        int d = sub.codePointBefore(2);
        System.out.println("Code point before char: " + d);
    }
}
```

What does this code write:

```
StringTokenizer stuff = new StringTokenizer( "abc,def,ghi" );
System.out.println( stuff.nextToken() );
```

What does this code write:

```
StringTokenizer stuff = new StringTokenizer( "abc,def,ghi", "," );
System.out.println( stuff.nextToken() );
```

What does this code write:

```
StringTokenizer stuff = new StringTokenizer( "abc+def+ghi", "+",
true );
System.out.println( stuff.nextToken() );
System.out.println( stuff.nextToken() );
```

What does this code write:

```
StringTokenizer stuff = new StringTokenizer( "abc def+ghi", "+");
System.out.println( stuff.nextToken() );
System.out.println( stuff.nextToken() );
```

What will be the value of str after the following statements are executed.

```
String str = "Dr. Decaffeinated";
StringBuffer strBuf = new StringBuffer(str.substring(6, 10) );
strBuf.setCharAt(1, 'o');
strBuf.append( 'e' );
str = strBuf.toString( );
```

public class Countcode {

```
    public static void main(String args[]) {
        StringBuffer sub = new StringBuffer("STRINGBUFFER");
        int c = sub.codePointCount(5, 10);
        System.out.println("Count: " + c);
        StringBuffer sub1 = new StringBuffer("STRINGBUILDER");
        c = sub1.codePointCount(1, 5);
        System.out.print("Count: " + c);
    }
}
```

public class Firstlast {

```
    public static void main(String args[]) {
         StringBuffer sub = new StringBuffer("I have two sisters,two
brothers and two uncles");
        int k;
        k = sub.indexOf("two");
        System.out.print(k + ",");
        k = sub.lastIndexOf("two");
        System.out.print(k);
    }
}
```

public class Offset {

```
    public static void main(String args[]) {
        StringBuffer ng = new StringBuffer("The list of files ");
        int i = ng.offsetByCodePoints(3, 8);
        System.out.print(i + ",");
        ng.append("contains sub files");
        ng.trimToSize();
        System.out.print(ng);
    }
}
```

What will be the output of the following program?

```
public class Countcode {
    public static void main(String args[]) {
        StringBuffer sub = new StringBuffer("STRINGBUFFER");
        int c = sub.codePointCount(5, 10, 6);
        System.out.println("Count: " + c);
        StringBuffer sub1 = new StringBuffer("STRINGBUILDER");
```

```
        c = sub1.codePointCount(1, 5, 4);
        System.out.print("Count: " + c);
    }
}

class StringBufferCodePointCount
{
    public static void main(String s[])
    {
        StringBuffer sb = new StringBuffer("core java");
        int start = 1;
        int end = 5;
         System.out.println("Number of code points in the portion of
the stringbuffer that are between start and end-1 is : " +
getCodePointCount(sb, start, end));

    }
public static int getCodePointCount(StringBuffer sb, int start, int
end) {

/*Write code here to find the number of code points in the portion
of the given stringbuffer that are between start and end-1. */
    }

}
```

Chapter 11: Arrays

```
class Test {
   public static void main(String args[]) {
     int arr[2];
     System.out.println(arr[0]);
     System.out.println(arr[1]);
   }
}

class Test {
   public static void main(String args[]) {
     int arr[] = new int[2];
     System.out.println(arr[0]);
     System.out.println(arr[1]);
   }
}

public class Main {
    public static void main(String args[]) {
        int arr[][] = new int[4][];
        arr[0] = new int[1];
        arr[1] = new int[2];
        arr[2] = new int[3];
        arr[3] = new int[4];

        int i, j, k = 0;
        for (i = 0; i < 4; i++) {
            for (j = 0; j < i + 1; j++) {
                arr[i][j] = k;
                k++;
            }
        }
        for (i = 0; i < 4; i++) {
            for (j = 0; j < i + 1; j++) {
                System.out.print(" " + arr[i][j]);
                k++;
            }
            System.out.println();
        }
    }
}

class Test
{
    public static void main (String[] args)
    {
        int arr1[] = {1, 2, 3};
        int arr2[] = {1, 2, 3};
        if (arr1 == arr2)
            System.out.println("Same");
        else
            System.out.println("Not same");
    }
}

class Test
{
    public static void main (String[] args)
    {
        int arr1[] = {1, 2, 3};
```

```
        int arr2[] = {1, 2, 3};
        if (arr1 == arr2)
            System.out.println("Same");
        else
            System.out.println("Not same");
    }
}

class Test
{
    public static void main (String[] args)
    {
        int arr1[] = {1, 2, 3};
        int arr2[] = {1, 2, 3};
        if (Arrays.equals(arr1, arr2))
            System.out.println("Same");
        else
            System.out.println("Not same");
    }
}

class Test
{
    public static void main (String[] args)
    {
        int arr1[] = {1, 2, 3};
        int arr2[] = {1, 2, 3};
        if (arr1.equals(arr2))
            System.out.println("Same");
        else
            System.out.println("Not same");
    }
}

public class Exercise9 {
public static void main(String[] args) {
   int[] my_array = {25, 14, 56, 15, 36, 56, 77, 18, 29, 49};
      // Insert an element in 3rd position of the array (index->2,
value->5)
      int Index_position = 2;
   int newValue    = 5;
                    System.out.println("Original          Array          :
"+Arrays.toString(my_array));
  for(int i=my_array.length-1; i > Index_position; i--){
    my_array[i] = my_array[i-1];
  }
  my_array[Index_position] = newValue;
  System.out.println("New Array: "+Arrays.toString(my_array));
}
}

public class Exercise7 {
public static void main(String[] args) {
   int[] my_array = {25, 14, 56, 15, 36, 56, 77, 18, 29, 49};

                              System.out.println("Original          Array          :
"+Arrays.toString(my_array));

  // Remove the second element (index->1, value->14) of the array
   int removeIndex = 1;

   for(int i = removeIndex; i < my_array.length -1; i++){
        my_array[i] = my_array[i + 1];
      }
// We cannot alter the size of an array , after the removal, the last
and second last element in the array will exist twice
          System.out.println("After removing the second element:
"+Arrays.toString(my_array));
 }
 }

public class Exercise2 {
public static void main(String[] args) {
int my_array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
int sum = 0;

for (int i : my_array)
    sum += i;
System.out.println("The sum is " + sum);
}
}

public class Exercise1 {
public static void main(String[] args){
    int[] my_array1 = {
          1789, 2035, 1899, 1456, 2013,
```

```
        1458, 2458, 1254, 1472, 2365,
        1456, 2165, 1457, 2456};
    String[] my_array2 = {
        "Java",
        "Python",
        "PHP",
        "C#",
        "C Programming",
        "C++"
      };
                System.out.println("Original numeric array :
"+Arrays.toString(my_array1));
    Arrays.sort(my_array1);
                 System.out.println("Sorted numeric array :
"+Arrays.toString(my_array1));
                  System.out.println("Original string array :
"+Arrays.toString(my_array2));
    Arrays.sort(my_array2);
                   System.out.println("Sorted string array :
"+Arrays.toString(my_array2));
    }
}

public class TechnicalInterviewTest {
    public static void main(String args[]) {
      int[][] test = new int[][]{
        {1, 1, 2, 2, 3, 4, 5},
        {1, 1, 1, 1, 1, 1, 1},
        {1, 2, 3, 4, 5, 6, 7},
        {1, 2, 1, 1, 1, 1, 1},};
      for (int[] input : test) {
            System.out.println("Array with Duplicates : " +
Arrays.toString(input));
            System.out.println("After removing duplicates : " +
Arrays.toString(removeDuplicates(input)));
      }
    }
                    public static int[] removeDuplicates(int[]
numbersWithDuplicates) {

      // Sorting array to bring duplicates together
      Arrays.sort(numbersWithDuplicates);
      int[] result = new int[numbersWithDuplicates.length];
      int previous = numbersWithDuplicates[0];
      result[0] = previous;

      for (int i = 1; i < numbersWithDuplicates.length; i++) {
        int ch = numbersWithDuplicates[i];

        if (previous != ch) {
          result[i] = ch;
        }
        previous = ch;
      }
      return result;

    }
}

public class JavaProgram
{
  public static void main(String args[])
  {
    int row, col, i, j;
    int arr[][] = new int[10][10];
    Scanner scan = new Scanner(System.in);
     System.out.print("Enter Number of Row for Array (max 10) :
");
    row = scan.nextInt();
      System.out.print("Enter Number of Column for Array (max
10) : ");
    col = scan.nextInt();

    System.out.print("Enter " +(row*col)+ " Array Elements : ");
    for(i=0; i<row; i++)
    {
      for(j=0; j<col; j++)
      {
        arr[i][j] = scan.nextInt();
      }
    }
    System.out.print("The Array is :\n");
    for(i=0; i<row; i++)
    {
      for(j=0; j<col; j++)
```

```
        {
           System.out.print(arr[i][j]+ "  ");
        }
        System.out.println();
     }
  }
}
```

public class JavaProgram

```
{
  public static void main(String args[])
  {
     int size, i, j, temp;
     int arr[] = new int[50];
     Scanner scan = new Scanner(System.in);

     System.out.print("Enter Array Size : ");
     size = scan.nextInt();

     System.out.print("Enter Array Elements : ");
     for(i=0; i<size; i++)
     {
        arr[i] = scan.nextInt();
     }
     j = i - 1;     // now j will point to the last element
     i = 0;         // and i will point to the first element
     while(i<j)
     {
        temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
        i++;
        j--;
     }
     System.out.print("Now the Reverse of Array is : \n");
     for(i=0; i<size; i++)
     {
        System.out.print(arr[i]+ "  ");
     }
  }
}
```

public class JavaProgram

```
{
  public static void main(String args[])
  {
     int size1, size2, size, i, j, k;
     int arr1[] = new int[50];
     int arr2[] = new int[50];
     int merge[] = new int[100];
     Scanner scan = new Scanner(System.in);
     System.out.print("Enter Array 1 Size : ");
     size1 = scan.nextInt();
     System.out.print("Enter Array 1 Elements : ");
     for(i=0; i<size1; i++)
     {
        arr1[i] = scan.nextInt();
     }
     System.out.print("Enter Array 2 Size : ");
     size2 = scan.nextInt();
     System.out.print("Enter Array 2 Elements : ");
     for(i=0; i<size2; i++)
     {
        arr2[i] = scan.nextInt();
     }
     System.out.print("Merging the Arrays...\n");
     for(i=0; i<size1; i++)
     {
        merge[i] = arr1[i];
     }
     size = size1 + size2;
     for(i=0, k=size1; k<size && i<size2; i++, k++)
     {
        merge[k] = arr2[i];
     }

     System.out.print("Now the New Array after Merging is :\n");
     for(i=0; i<size; i++)
     {
        System.out.print(merge[i] + "  ");
     }
  }
}
```

What will be the output of the following program?

```
public class College {
```

```
    public static void main(String[] args) {
        Student[] student = new Student[2];
        student[0] = new Student();
        student[0].name = "Khan";
        student[0] = new Student();
        student[0].name = "Kittu";
        student[1] = new Student();
        student[1].name = "Munna";
        for (Student element : student) {
            System.out.print(element.name + " ~ ");
        }
    }
}
class Student {
    String name;
}
```

Chapter 12: Garbage Collection

```
void start() {
    A a = new A();
    B b = new B();
    a.s(b);
    b = null; /* Line 5 */
    a = null;  /* Line 6 */
    System.out.println("start completed"); /* Line 7 */
}
```

When is the B object, created in line 3, eligible for garbage collection?

```
class HappyGarbage01
{
    public static void main(String args[])
    {
        HappyGarbage01 h = new HappyGarbage01();
        h.methodA(); /* Line 6 */
    }
    Object methodA()
    {
        Object obj1 = new Object();
        Object [] obj2 = new Object[1];
        obj2[0] = obj1;
        obj1 = null;
        return obj2[0];
    }
}
```

Where will be the most chance of the garbage collector being invoked?

```
class Bar { }
class Test
{
    Bar doBar()
    {
        Bar b = new Bar(); /* Line 6 */
        return b; /* Line 7 */
    }
    public static void main (String args[])
    {
        Test t = new Test();  /* Line 11 */
        Bar newBar = t.doBar();  /* Line 12 */
        System.out.println("newBar");
        newBar = new Bar(); /* Line 14 */
        System.out.println("finishing"); /* Line 15 */
    }
}
```

At what point is the Bar object, created on line 6, eligible for garbage collection?

```
public class Test
{
    public static void main(String[] args) throws
InterruptedException
    {
        Test t = new Test();
        // making t eligible for garbage collection
        t = null;
        // calling garbage collector
        System.gc();
        // waiting for gc to complete
        Thread.sleep(1000);
        System.out.println("end main");
    }
     @Override
    protected void finalize()
    {
        System.out.println("finalize method called");
```

```
        System.out.println(10/0);
    }

}

public class Test
{
    static Test t ;
    static int count =0;
    public static void main(String[] args) throws
InterruptedException
    {
        Test t1 = new Test();
        // making t1 eligible for garbage collection
        t1 = null; // line 12
        // calling garbage collector
        System.gc(); // line 15
        // waiting for gc to complete
        Thread.sleep(1000);
        // making t eligible for garbage collection,
        t = null; // line 21
        // calling garbage collector
        System.gc(); // line 24

        // waiting for gc to complete
        Thread.sleep(1000);
        System.out.println("finalize method called "+count+"
times");
    }
    @Override
    protected void finalize()
    {
        count++;
        t = this; // line 38
    }
}
```

public class Test

```
{
    public static void main(String[] args)
    {
        // How many objects are eligible for
        // garbage collection after this line?
        m1();  // Line 5
    }

    static void m1()
    {
        Test t1 = new Test();
        Test t2 = new Test();
    }
}
```

How many objects are eligible for garbage collection after execution of line 5 ?

```
public class Test
{
    public static void main(String [] args)
    {
        Test t1 = new Test();
        Test t2 = m1(t1); // line 6
        Test t3 = new Test();
        t2 = t3; // line 8

    }
    static Test m1(Test temp)
    {
        temp = new Test();
        return temp;
    }
}
```

How many objects are eligible for garbage collection after execution of line 8?

```
class X2
{
    public X2 x;
    public static void main(String [] args)
    {
        X2 x2 = new X2();  /* Line 6 */
        X2 x3 = new X2();  /* Line 7 */
        x2.x = x3;
        x3.x = x2;
        x2 = new X2();
        x3 = x2; /* Line 11 */
        doComplexStuff();
```

```
    }
}
```

after line 11 runs, how many objects are eligible for garbage collection?

```
public Object m()
{
   Object o = new Float(3.14F);
   Object [] oa = new Object[l];
   oa[0] = o; /* Line 5 */
   o = null;  /* Line 6 */
   oa[0] = null; /* Line 7 */
   return o; /* Line 8 */
}
```

When is the Float object, created in line 3, eligible for garbage collection?

```
public class X
{
   public static void main(String [] args)
   {
      X x = new X();
      X x2 = m1(x); /* Line 6 */
      X x4 = new X();
      x2 = x4; /* Line 8 */
      doComplexStuff();
   }
   static X m1(X mx)
   {
      mx = new X();
      return mx;
   }
}
```

After line 8 runs. how many objects are eligible for garbage collection?

```
class Test
{
   private Demo d;
   void start()
   {
      d = new Demo();
      this.takeDemo(d); /* Line 7 */
   } /* Line 8 */
   void takeDemo(Demo demo)
   {
      demo = null;
      demo = new Demo();
   }
}
```

When is the Demo object eligible for garbage collection?

References:
www.proprofs.com, meritcampus.com, www.geeksforgeeks.org, www.f5java.com, dzone.com, www.examveda.com, beingzero.in/, scjptest.com, www.gocertify.com, web.cs.iastate.edu, www.sanfoundry.com, https://codingpuzzles.com , www.quizover.com, www.developer.com, www.meritcampus.com, https://docs.oracle.com, www.journaldev.com, http://www.mhhe.com, www.w3resource.com, javarevisited.blogspot.in, codescracker.com, www.indiabix.com, resume.mcalglobal.com,