



# CODERS LODGE

## JAVA PROGRAMMING OUTPUT BASED QUESTIONS

Kindly read the instructions carefully

1. All these questions are important for examination and interview point of view, so practice them well.
2. If you have any doubt or facing any problem regarding these questions you can mail us at [coderslodgeofficial@gmail.com](mailto:coderslodgeofficial@gmail.com) or drop a message in our WhatsApp or telegram group.
3. If you want to support us, give your valuable feedback so that next time we can improve while interacting with you.
4. ***Reminder***-Practice all questions well it will build your concept clear and you can easily score good in your exams.

### Connect with us

- If you want to join our **WhatsApp** community then mail us at: - [coderslodgeofficial@gmail.com](mailto:coderslodgeofficial@gmail.com)
- Join our **Telegram** group: - <https://t.me/coderslodgeofficial>
- Like our **Facebook** page: - <https://www.facebook.com/coderslodge>
- Follow us on **Instagram**:- <https://www.instagram.com/coderslodge/>
- Follow us on **Twitter**: - <https://twitter.com/CodersLodge>
- Follow us on **LinkedIn**:- <https://www.linkedin.com/company/coderslodge>



1. What will be the output of the following Java code?

```
class increment {  
    public static void main(String args[])  
    {  
        int g = 3;  
        System.out.print(++g * 8);  
    }  
}
```

- a) 32
- b) 33
- c) 24
- d) 25

Ans: a

2. What will be the output of the following Java code?

```
class output {  
    public static void main(String args[])  
    {  
        double a, b,c;  
        a = 3.0/0;  
        b = 0/4.0;  
        c=0/0.0;
```



# CODERS LODGE

```
System.out.println(a);  
System.out.println(b);  
System.out.println(c);  
}  
}
```

- a) NaN
- b) Infinity
- c) 0.0
- d) all of the mentioned

Ans: a

### 3. What will be the output of the following Java program?

```
class variable_scope  
{  
    public static void main(String args[])  
    {  
        int x;  
        x = 5;  
        {  
            int y = 6;  
            System.out.print(x + " " + y);  
        }  
        System.out.println(x + " " + y);  
    }  
}
```



```
    }  
}
```

- a) Compilation error
- b) Runtime error
- c) 5 6 5 6
- d) 5 6 5

Ans: a

#### 4. What will be the output of the following Java program?

```
class evaluate  
{  
    public static void main(String args[])  
    {  
        int arr[] = new int[] {0 , 1, 2, 3, 4, 5, 6, 7, 8, 9};  
        int n = 6;  
        n = arr[arr[n] / 2];  
        System.out.println(arr[n] / 2);  
    }  
}
```

- a) 2
- b) 1
- c) 4
- d) 0

Ans: b



# CODERS LODGE

## 5. Predict the output of the following Java program.

```
package main;

class Base {
    public void Print()
    {
        System.out.println("Base");
    }
}

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

class Main {
    public static void DoPrint(Base o)
    {
        o.Print();
    }

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



# CODERS LODGE

```
Base x = new Base();  
Base y = new Derived();  
Derived z = new Derived();  
DoPrint(x);  
DoPrint(y);  
DoPrint(z);  
}  
}
```

Ans: Base

Derived

Derived

## 6. Predict the output.

```
package main;  
// filename Main.java  
class Point {  
    protected int x, y;  
  
    public Point(int _x, int _y)  
    {  
        x = _x;  
        y = _y;  
    }  
}  
public class Main {  
    public static void main(String args[]){
```



# CODERS LODGE

```
        Point p = new Point();  
        System.out.println("x = " + p.x + ", y = " + p.y);  
    }  
}
```

Ans: Compiler Error

## 7. What will be the output of the following Java program?

```
class leftshift_operator  
{  
    public static void main(String args[])  
    {  
        byte x = 64;  
        int i;  
        byte y;  
        i = x << 2;  
        y = (byte) (x << 2)  
        System.out.print(i + " " + y);  
    }  
}
```

- a) 0 256
- b) 0 64
- c) 256 0
- d) 64 0

Ans: c



## 8. Predict the output.

*// filename: Test.java*

```
class Test {  
  
    // Declaring and initializing integer variable  
    int x = 10;  
  
    // Main driver method  
    public static void main(String[] args)  
    {  
        // Creating an object of class inside main()  
        Test t = new Test();  
  
        // Printing the value inside the object by  
        // above created object  
        System.out.println(t.x);  
    }  
}
```

Ans: 10

## 9. Predict the output.

*// filename: Test.java*

*// Main class*

```
class Test {  
    // Declaring and initializing variables  
    int y = 2;  
    int x = y + 2;
```





# CODERS LODGE

```
// main driver method
public static void main(String[] args)
{

    // Creating an object of class inside main() method
    Test m = new Test();

    // Printing the value of x and y
    // using above object created
    System.out.println("x = " + m.x + ", y = " + m.y);

}
}
```

Ans: x = 4, y = 2

10. What will be the output of the following Java program?

```
class box
{
    int width;
    int height;
    int length;
}

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



# CODERS LODGE

```
{  
    box obj = new box();  
    obj.width = 10;  
    obj.height = 2;  
    obj.length = 10;  
    int y = obj.width * obj.height * obj.length;  
    System.out.print(y);  
}  
}
```

- a) 100
- b) 400
- c) 200
- d) 12

Ans: c

## 11. Predict the output.

// file name: Main.java

```
class Base {  
    protected void foo() {}  
}  
class Derived extends Base {  
    void foo() {}  
}  
public class Main {
```



# CODERS LODGE

```
public static void main(String args[]) {  
    Derived d = new Derived();  
    d.foo();  
}  
}
```

Ans: Compiler Error

## 12. Predict the output.

```
// Main.java  
public class Main  
{  
    public static void main(String args[])  
    {  
        String s1 = "abc";  
        String s2 = s1;  
        s1 += "d";  
        System.out.println(s1 + " " + s2 + " " + (s1 == s2));  
  
        StringBuffer sb1 = new StringBuffer("abc");  
        StringBuffer sb2 = sb1;  
        sb1.append("d");  
        System.out.println(sb1 + " " + sb2 + " " + (sb1 ==  
sb2));  
    }  
} //end class
```

Ans: abcd abc false

abcd abcd true



13. What will be the output of the following Java program?

```
class Alligator
{
    public static void main(String[] args)
    {
        int []x[] = {{1,2}, {3,4,5}, {6,7,8,9}};
        int [][]y = x;
        System.out.println(y[2][1]);
    }
}
```

- a) Compilation Error
- b) 2
- c) 3
- d) 7

Ans: d

14. What will be the output of the following Java code?

```
class A
{
    int i;
    void display()
    {
        System.out.println(i);
    }
}
```



# CODERS LODGE

```
    }  
}  
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();  
    }  
}
```



- a) 1
- b) 2
- c) 0
- d) Error

Ans: b

**15. Predict the output.**

```
class First
{
    public First() { System.out.println("a"); }
}

class Second extends First
{
    public Second() { System.out.println("b"); }
}

class Third extends Second
{
    public Third() { System.out.println("c"); }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Third c = new Third();
    }
}
```



# CODERS LODGE

Ans: a  
b  
c

**16. Predict the output.**

```
class First
{
    int i = 10;

    public First(int j)
    {
        System.out.println(i);
        this.i = j * 10;
    }
}

class Second extends First
{
    public Second(int j)
    {
        super(j);
        System.out.println(i);
        this.i = j * 20;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Second n = new Second(20);
    }
}
```



```
        System.out.println(n.i);
    }
}
```

Ans: 10  
200  
400

**17. Predict the output.**

```
class ThreadEx extends Thread
{
    public void run()
    {
        System.out.print("Hello...");
    }
    public static void main(String args[])
    {
        ThreadEx T1 = new ThreadEx();
        T1.start();
        T1.stop();
        T1.start();
    }
}
```

Ans: Run Time Exception

**18. What will be the output of the following Java code?**

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





# CODERS LODGE

```
{  
    char chars[] = {'a', 'b', 'c'};  
    String s = new String(chars);  
    System.out.println(s);  
}  
}
```

- a) abc
- b) a
- c) b
- d) c

Ans: a

**19. Predict the output.**

```
public class Calculator  
{  
    int num = 100;  
    public void calc(int num) { this.num = num * 10; }  
    public void printNum() { System.out.println(num); }  
  
    public static void main(String[] args)  
    {  
        Calculator obj = new Calculator();  
        obj.calc(2);  
    }  
}
```



# CODERS LODGE

```
        obj.printNum();  
    }  
}
```

- a) 20
- b) 100
- c) 1000
- d) 2

Ans: a

**20. What will be the output of the following Java code?**

```
class output {  
    public static void main(String args[])  
    {  
        String c = "Hello i love java";  
        boolean var;  
        var = c.startsWith("hello");  
        System.out.println(var);  
    }  
}
```

- a) 0
- b) true
- c) 1
- d) false

Ans: d



21. What will be the output of the following Java program?

```
class output
{
    public static void main(String args[])
    {
        StringBuffer s1 = new StringBuffer("Hello");
        StringBuffer s2 = s1.reverse();
        System.out.println(s2);
    }
}
```

- a) HelloolleH
- b) olleHHello
- c) Hello
- d) olleH

Ans: d

22. Predict the output.

```
class Alpha
{
    public String type = "a ";
    public Alpha() { System.out.print("alpha "); }
}

public class Beta extends Alpha
{
```



# CODERS LODGE

```
public Beta() { System.out.print("beta "); }

void go()
{
    type = "b ";
    System.out.print(this.type + super.type);
}

public static void main(String[] args)
{
    new Beta().go();
}
}
```

- a) alpha beta b b
- b) alpha beta a b
- c) beta alpha b b
- d) beta alpha a b

Ans: a

## 23. Predict the output.

```
public class Test
{
    public static void main(String[] args)
    {
        StringBuilder s1 = new StringBuilder("Java");
        String s2 = "Love";
        s1.append(s2);
        s1.substring(4);
    }
}
```



# CODERS LODGE

```
        int foundAt = s1.indexOf(s2);  
        System.out.println(foundAt);  
    }  
}
```

- a) -1
- b) 3
- c) 4
- d) A StringIndexOutOfBoundsException is thrown at runtime.

Ans: c

**24. What will be the output of the following Java program?**

```
class Output  
{  
    public static void main(String args[])  
    {  
        double x = 2.0;  
        double y = 3.0;  
        double z = Math.pow( x, y );  
        System.out.print(z);  
    }  
}
```

- a) 9.0
- b) 8.0



c) 4.0

d) 2.0

Ans: b

**25. What will be the output of the following Java code?**

```
class Output
{
    public static void main(String args[])
    {
        int a = Character.MIN_VALUE;
        System.out.print((char)a);
    }
}
```

a) @

b) Space

c) <

d) !

Ans: b

**26. Predict the output.**

```
class superClass
{
    final public int calc(int a, int b)
    {
        return 0;
    }
}
```



# CODERS LODGE

```
}  
class subClass extends superClass  
{  
    public int calc(int a, int b)  
    {  
        return 1;  
    }  
}  
public class Gfg  
{  
    public static void main(String args[])  
    {  
        subClass get = new subClass();  
        System.out.println("x = " + get.calc(0, 1));  
    }  
}
```

Ans: Compilation fails

27. What will be the output of the following Java code snippet?

```
import java.util.*;  
class ArrayList  
{  
    public static void main(String args[])  
    {  
        ArrayList obj = new ArrayList();  
        obj.add("A");  
        obj.add("B");  
    }  
}
```



# CODERS LODGE

```
obj.add("C");  
obj.add(1, "D");  
System.out.println(obj);  
}  
}
```

- a) [A, D, C]
- b) [A, B, C]
- c) [A, B, C, D]
- d) [A, D, B, C]

Ans: d

**28. What will be the output of the following Java code?**

```
class multithreaded_programing  
{  
    public static void main(String args[])  
    {  
        Thread t = Thread.currentThread();  
        t.setName("New Thread");  
        System.out.println(t);  
    }  
}
```

- a) Thread[main,5,main]
- b) Thread[New Thread,5,main]





# CODERS LODGE

- c) Thread[5,main]
- d) Thread[New Thread,5]

Ans: b

**29. What will be the output of the following Java program?**

```
class output
{
    public static void main(String args[])
    {
        StringBuffer s1 = new StringBuffer("Hello");
        StringBuffer s2 = s1.reverse();
        System.out.println(s2);
    }
}
```

- a) olleH
- b) olleHHello
- c) Hello
- d) HelloolleH

Ans: a

**30. What will be the output of the following Java code?**

```
class newthread extends Thread
{
    Thread t;
```



# CODERS LODGE

```
newthread()
{
    t1 = new Thread(this,"Thread_1");
    t2 = new Thread(this,"Thread_2");
    t1.start();
    t2.start();
}
public void run()
{
    t2.setPriority(Thread.MAX_PRIORITY);
    System.out.print(t1.equals(t2));
}
}
class multithreaded_programing
{
    public static void main(String args[])
    {
        new newthread();
    }
}
```



# CODERS LODGE

- a) true true
- b) false false
- c) true
- d) false

Ans: b

### 31. Predict the output.

```
public class Base
```

```
{
```

```
private int data;
```

```
public Base()
```

```
{
```

```
data = 5;
```

```
}
```

```
public int getData()
```

```
{
```

```
return this.data;
```

```
}
```

```
}
```

```
class Derived extends Base
```

CODERS LODGE



# CODERS LODGE

```
{  
    private int data;  
    public Derived()  
    {  
        data = 6;  
    }  
    private int getData()  
    {  
        return data;  
    }  
  
    public static void main(String[] args)  
    {  
        Derived myData = new Derived();  
        System.out.println(myData.getData());  
    }  
}
```

- a) 6
- b) 5
- c) Compile time error
- d) Run time error

Ans: c



# CODERS LODGE

32. What will be the output of the following Java code?

```
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
{
```



# CODERS LODGE

```
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);
}
}
```

- a) 6.4 6.4
- b) 6 6
- c) 4 6.4
- d) 6.4 6

Ans: c

### 33. Predict the output.

```
import java.io.IOException;
import java.util.EmptyStackException;

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



# CODERS LODGE

```
    {  
        System.out.printf("%d", 1);  
        throw(new Exception());  
    }  
    catch(IOException e)  
    {  
        System.out.printf("%d", 2);  
    }  
    catch(EmptyStackException e)  
    {  
        System.out.printf("%d", 3);  
    }  
    catch(Exception e)  
    {  
        System.out.printf("%d", 4);  
    }  
    finally  
    {  
        System.out.printf("%d", 5);  
    }  
}
```

- a) 12345
- b) 15
- c) 135
- d) 145

Ans: d



34. What will be the output of the following Java program?

```
class string_class
{
    public static void main(String args[])
    {
        String obj = "I LIKE JAVA";
        System.out.println(obj.length());
    }
}
```

- a) 11
- b) 12
- c) 10
- d) 9

Ans: a

35. Predict the output.

```
public class Test implements Runnable
{
    public void run()
    {
        System.out.printf("%d",3);
    }
    public static void main(String[] args) throws
    InterruptedException
    {
```





# CODERS LODGE

```
Thread thread = new Thread(new Test());
thread.start();
System.out.printf("%d",1);
thread.join();
System.out.printf("%d",2);
}
}
```

- a) 123
- b) 213
- c) 132
- d) 321

Ans: c

**36. Predict the output.**

```
import java.io.*;
public class Test
{
    public void display() throws IOException
    {
        System.out.println("Test");
    }
}

class Derived extends Test
{
    public void display() throws IOException
```



# CODERS LODGE

```
{  
    System.out.println("Derived");  
}  
public static void main(String[] args) throws IOException  
{  
    Derived object = new Derived();  
    object.display();  
}  
}
```

- a) Test
- b) Derived
- c) Compilation error
- d) Runtime error

Ans: b

### 37. Predict the output.

```
class Temp  
{  
    private Temp(int data)  
    {  
        System.out.printf(" Constructor called ");  
    }  
    protected static Temp create(int data)  
    {  
        Temp obj = new Temp(data);  
        return obj;  
    }  
    public void myMethod()  
}
```



# CODERS LODGE

```
    {  
        System.out.printf(" Method called ");  
    }  
}
```

```
public class Test  
{  
    public static void main(String[] args)  
    {  
        Temp obj = Temp.create(20);  
        obj.myMethod();  
    }  
}
```

- a) Constructor called Method called
- b) Compilation error
- c) Runtime error
- d) None of the above

Ans: a

### 38. Predict the output.

```
class Base  
{  
    public static String s = " Super Class ";  
    public Base()  
    {  
        System.out.printf("1");  
    }  
}  
public class Derived extends Base  
{
```



# CODERS LODGE

```
public Derived()
{
    System.out.printf("2");
    super();
}

public static void main(String[] args)
{
    Derived obj = new Derived();
    System.out.printf(s);
}
}
```

- a) 21 Super Class
- b) Super Class 21
- c) Compilation error
- d) 12 Super Class

Ans: c

## 39. Predict the output.

```
public class Outer
{
    private static int data = 10;
    private static int LocalClass()
    {
        class Inner
        {
            public int data = 20;
            private int getData()

```



# CODERS LODGE

```
        {  
            return data;  
        }  
};  
Inner inner = new Inner();  
return inner.getData();  
}  
  
public static void main(String[] args)  
{  
    System.out.println(data * LocalClass());  
}  
}
```

- a) Compilation error
- b) Runtime Error
- c) 200
- d) None of the above

Ans: c

CODERS LODGE

#### 40. Predict the output.

```
// Java code for thread creation by extending the Thread class  
class MultithreadingDemo extends Thread {  
    public void run()  
    {  
        try {  
            // Displaying the thread that is running  
            System.out.println(  

```



# CODERS LODGE

```
        "Thread " +  
Thread.currentThread().getId()  
        + " is running");  
    }  
    catch (Exception e) {  
        // Throwing an exception  
        System.out.println("Exception is caught");  
    }  
}  
}  
  
// Main Class  
public class Multithread {  
    public static void main(String[] args)  
    {  
        int n = 8; // Number of threads  
        for (int i = 0; i < n; i++) {  
            MultithreadingDemo object  
                = new MultithreadingDemo();  
            object.start();  
        }  
    }  
}
```

Ans: Thread 15 is running  
Thread 14 is running  
Thread 16 is running  
Thread 12 is running



# CODERS LODGE

Thread 11 is running

Thread 13 is running

Thread 18 is running

Thread 17 is running



CODERS LODGE