

# OOP Questions in Java

Collected by:  
Omar AbdulRahman

```
class Test {  
    public static void main(String[] args) {  
        for(int i = 0; 1; i++) {  
            System.out.println("Hello");  
            break;  
        }  
    }  
}
```

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

```
public class A {  
    public static void main(String[] args)  
    {  
        if (true)  
            break;  
    }  
}
```

Choices:

- a) Nothing
- b) Error

```
public class B {  
    public static void main(String[] args)  
    {  
        System.out.println('j' + 'a' + 'v' +  
'a');  
    }  
}
```

Choices:

a) java

b) Something else (Other than simple concatenation)

```
public class C {  
    public static void main(String[] args)  
    {  
        int $_ = 5;  
    }  
}
```

Choices:

- a) Nothing
- b) Error

```
public class D{
    public static void main(String[] arr){

    }
    public static void main(String arr){

    }
}
```

Choices:

- a) Nothing
- b) Error

```
class Derived
{
    protected final void getDetails()
    {
        System.out.println("Derived class");
    }
}

public class Test extends Derived
{
    protected final void getDetails()
    {
        System.out.println("Test class");
    }
    public static void main(String[] args)
    {
        Derived obj = new Derived();
        obj.getDetails();
    }
}
```

## Choices:

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



```
class Derived
{
    public void getDetails(String temp)
    {
        System.out.println("Derived class " + temp);
    }
}

public class Test extends Derived
{
    public int getDetails(String temp)
    {
        System.out.println("Test class " + temp);
        return 0;
    }
    public static void main(String[] args)
    {
        Test obj = new Test();
        obj.getDetails("GFG");
    }
}
```

## Choices:

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

```
class Derived
{
    public void getDetails()
    {
        System.out.println("Derived class");
    }
}

public class Test extends Derived
{
    protected void getDetails()
    {
        System.out.println("Test class");
    }
    public static void main(String[] args)
    {
        Derived obj = new Test(); // line xyz
        obj.getDetails();
    }
}
```

## Choices:

- a) Test class
- b) Compilation error due to line xyz
- c) Derived class
- d) Compilation error due to access modifier

```
class Derived
{
    public void getDetails()
    {
        System.out.printf("Derived class ");
    }
}

public class Test extends Derived
{
    public void getDetails()
    {
        System.out.printf("Test class ");
        super.getDetails();
    }
    public static void main(String[] args)
    {
        Derived obj = new Test();
        obj.getDetails();
    }
}
```

## Choices:

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

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

```
public class Main
{
    public static void myFynction(String s)
    {
        System.out.println("String");
    }
    public static void myFynction(Object o)
    {
        System.out.println("Object");
    }

    public static void main(String args[])
    {
        myFynction(null);
    }
}
```

```
public class Main
{
    public static void gfg(String s)
    {
        System.out.println("String");
    }
    public static void gfg(Object o)
    {
        System.out.println("Object");
    }
    public static void gfg(Integer i)
    {
        System.out.println("Integer");
    }

    public static void main(String args[])
    {
        gfg(null); //line 19
    }
}
```

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

```
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();
    }
}
```



```
class Writer
{
    public static void write()
    {
        System.out.println("Writing...");
    }
}
class Author extends Writer
{
    public static void write()
    {
        System.out.println("Writing book");
    }
}

public class Programmer extends Author
{
    public static void write()
    {
        System.out.println("Writing code");
    }

    public static void main(String[] args)
    {
        Author a = new Programmer();
        a.write();
    }
}
```

- A) Writing...
- B) Writing book
- C) Writing code
- D) Compilation fails

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

```
public class Base
{
    private int data;

    public Base()
    {
        data = 5;
    }

    public int getData()
    {
        return this.data;
    }
}
```

```
class Derived extends Base
{
    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());
    }
}
```

Choices:

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

```
public class Test
{
    public static void main(String[] args)
    {
        try
        {
            System.out.printf("1");
            int sum = 9 / 0;
            System.out.printf("2");
        }
        catch(ArithmeticException e)
        {
            System.out.printf("3");
        }
        catch(Exception e)
        {
            System.out.printf("4");
        }
        finally
        {
            System.out.printf("5");
        }
    }
}
```

Choices:

- a) 1325
- b) 1345
- c) 1342
- d) 135

```

public class Test
{
    private void m1()
    {
        m2();
        System.out.printf("1");
    }
    private void m2()
    {
        m3();
        System.out.printf("2");
    }
    private void m3()
    {
        System.out.printf("3");
        try
        {
            int sum = 4/0;
            System.out.printf("4");
        }
        catch(ArithmeticException e)
        {
            System.out.printf("5");
        }

        System.out.printf("7");
    }
    public static void main(String[] args)
    {
        Test obj = new Test();
        obj.m1();
    }
}

```

Choices:

- a) 35721
- b) 354721
- c) 3521
- d) 35

```
public class Test
{
    public static void main(String[] args)
    {
        try
        {
            System.out.printf("1");
            int data = 5 / 0;
        }
        catch(ArithmeticException e)
        {
            System.out.printf("2");
            System.exit(0);
        }
        finally
        {
            System.out.printf("3");
        }
        System.out.printf("4");
    }
}
```

Choices:

- a) 12
- b) 1234
- c) 124
- d) 123

```
class Helper
{
    private int data;
    private Helper()
    {
        data = 5;
    }
}
public class Test
{
    public static void main(String[] args)
    {
        Helper help = new Helper();
        System.out.println(help.data);
    }
}
```

Choices:

- a) Compilation error
- b) 5
- c) Runtime error
- d) None of these

```
public class Test
{
    private static float temp()
    {
        public static float sum = 21;
        return(--(sum));
    }
    public static void main(String[] args)
    {
        Test test = new Test();
        System.out.println(test.temp());
    }
}
```

Choices:

- a) 21
- b) 20
- c) Compilation error
- d) Runtime error



```
public class Test
{
    public static void main(String[] args)
    {
        int temp = 40;
        if(temp == 30 && temp/0 == 4)
        {
            System.out.println(1);
        }
        else
        {
            System.out.println(2);
        }
    }
}
```

Choices:

- a) 1
- b) 2
- c) Runtime Exception of java.lang.ArithmeticException
- d) Compilation error due to divisibility by 0

```
public class Test
{
    public static void main(String[] args)
    {
        int value = 554;
        String var = (String)value; //line 1
        String temp = "123";
        int data = (int)temp; //line 2
        System.out.println(data + var);
    }
}
```

Choices:

- a) 677
- b) Compilation error due to line 1
- c) Compilation error due to line 2
- d) Compilation error due to line 1 and line 2

```
class B
{
    public static String sing()
    {
        return "la";
    }
}

public class A extends B
{
    public static String sing()
    {
        return "fa";
    }
    public static void main(String[] args)
    {
        A a = new A();
        B b = new A();
        System.out.println(
            a.sing() + " " + b.sing()
        );
    }
}
```

```
class Base
{
    final public void show()
    {
        System.out.println("called from Base")
    }
;
}
class Derived extends Base
{
    public void show()
    {
        System.out.println(
            " called from Derived"
        );
    }
}
class Main
{
    public static void main(String[] args)
    {
        Base b = new Derived();
        b.show();
    }
}
```

```
class Example {
    private int x;
    public static void main(String args[])
    {
        Example obj = new Example();
    }
    public void Example(int x)
    {
        System.out.println(x);
    }
}
```

Choices:

- A. 0
- B. Garbage value
- C. Compile time error
- D. No output : Blank Screen

```
public class Example {
    int x = 10;
    public static void main(String args[])
    {
        Example obj;
        System.out.println(obj.x);
    }
}
```

Choices:

- A. 10
- B. 0
- C. Compile time error
- D. Run time error

```
class Test {
    public static void main(String[] args) {
        boolean b = true;
        if (b = false) {
            System.out.println("HELLO");
        } else {
            System.out.println("BYE");
        }
    }
}
```

Choices:

1.HELLO

2.BYE

3.Compile time error: re- initialization

4.No output

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

Choices:

1. HIII
2. HIII(infinitely)
3. BYE
4. Compile time error



```
public class Test {  
    public static void main(String[] args)  
    {  
        int x = 10;  
        if (x) {  
            System.out.println("HELLO GEEKS");  
        } else {  
            System.out.println("BYE");  
        }  
    }  
}
```

## Choices:

1. HELLO GEEKS
2. Compile time error
3. Runtime Error
4. BYE

```
public class Test {  
    public static void main(String[] args)  
    {  
        int x = 10;  
        if (x)  
            System.out.println("HELLO GEEKS");  
        System.out.println("WELCOME");  
  
        else  
        {  
            System.out.println("BYE");  
        }  
    }  
}
```

## Choices:

1. HELLO GEEKS  
WELCOME
2. HELLO GEEKS
3. BYE

```
public class Test {
public static void main(String[] args)
{
    int x = 10, y = 20;
    if (x < y)
        int a = 10;
    else {
        System.out.println("BYE");
    }
}
}
```

Choices:

1. 10
2. BYE
3. NO output
4. Compile time error

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

## Choices:

1. No Output
2. Compile time error
3. Runtime error
4. Runtime Exception

```
class MainClass {
    public static void main(String[] args)
    {
        int x = 10;
        int y = 20;
        switch (x) {
            case 10:
                System.out.println("HELLO");
                break;
            case y:
                System.out.println("GEEKS");
                break;
        }
    }
}
```

Choices:

1. HELLO
2. No Output
3. GEEKS
4. Compile time error

```
class MainClass {
    public static void main(String[] args)
    {
        int x = 10;
        final int y = 20;
        switch (x) {
            case 10:
                System.out.println("HELLO");
                break;
            case y:
                System.out.println("GEEKS");
                break;
        }
    }
}
```

## Choices:

1. GEEKS
2. Compile time error
3. HELLO
4. NO Output

```
public class Test {  
    public void main(String[] args) {  
        int x = 10*20-20;  
        System.out.println(x);  
    }  
}
```

## Choices:

- A. Runtime Error
- B. Prints 180
- C. Prints 0
- D. Compile-time error.

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

Choices:

- A. abcabc
- B. abc
- C. null



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

## Choices:

- A. s1 == s2 is:true
- B. false
- C. s1 == s2 is:false
- D. true

```
String x = new String("xyz");  
String y = "abc";  
x = x + y;
```

How many String objects were created?

Choices:

- A. 2
- B. 3
- C. 4
- D. 5

```
class Test
{
    public static void main (String args[])
    {
        System.out.println(
            10 + 20 + "Javatpoint"
        );
        System.out.println(
            "Javatpoint" + 10 + 20
        );
    }
}
```