

✗ `public class MyClass {` *
 `private int number = 10;`

 `public void display() {`
 `System.out.println(number);`
 `}`
`}`

 `public class Main {`
 `public static void main(String[] args) {`
 `MyClass obj = new MyClass();`
 `System.out.println(obj.number);`
 `}`
 `}`

0/1

- ☐ 10
- ☒ Error: Access Denied
- ☐ 0
- ☐ Compilation Error

✗

Correct answer

- ☒ Compilation Error

✗ Which of the following is a generic class in Java? *

0/1

- ☒ `HashMap`
- ☐ `Integer`
- ☐ `String`
- ☐ `ArrayList`

✗

Correct answer

- ☒ `ArrayList`



✓ public class Test * 1/1

```
{
    public int a=0;
    class innerClass
    {
        public int a=1;
        void innermethod(int x)
        {
            System.out.println("value of x = " + x);
            System.out.println("value of this.x = " + this.x);
            System.out.println("value of Test.this.x = " + Test.T=this.x);
        }
    }
}

public static void main( String args[] )
{
    Test t=new Test();
    Test.innerClass im=t.new innerClass();
    im.innermethod(55);
}
```

- ☐ value of x = 55, value of this.x = 0, value of Test.this.x = 1
- ☐ value of x = 1, value of this.x = 0, value of Test.this.x = 55
- ☒ value of x = 55, value of this.x = 1, value of Test.this.x = 0 ✓
- ☐ value of x = 0, value of this.x = 55, value of Test.this.x = 1

✓ How to create object of the inner class? * 1/1

- ☒ OuterClass.InnerClass innerObject = outerObject.new InnerClass(); ✓
- ☐ OuterClass.InnerClass innerObject = new InnerClass();
- ☐ InnerClass innerObject = outerObject.new InnerClass();
- ☐ OuterClass.InnerClass = outerObject.new InnerClass();

✓ What happens if you try to unbox a 'null' reference in Java? *

1/1

- ☐ It throws a `ClassCastException`
- ☐ It returns `null`
- ☐ It returns the default value for the wrapper class
- ☒ It throws a `NullPointerException`



✓ class One{ *

1/1

```
    public static void print(){  
        System.out.println("1");  
    }  
}
```

```
class Two extends One{  
    public static void print(){  
        System.out.println("2");  
    }  
}
```

```
public class Test{  
    public static void main(String args[]){  
        One one = new Two();  
        one.print();  
    }  
}
```

- ☐ 2
- ☒ 1
- ☐ Compile-time error
- ☐ Run-time error



✓ Which of the following exception must be either caught or declared to be thrown in Java? *1/1

- ☐ NullPointerException
- ☐ ArrayIndexOutOfBoundsException
- ☒ FileNotFoundException
- ☐ ArithmeticException



✓ Which access modifier restricts access of a class member to only its subclasses and classes in the same package? *1/1

- ☐ Private
- ☐ Default
- ☐ Public
- ☒ Protected



✓ public class Test

*

1/1

```
{
    public static void main(String[] args)
    {
        try
        {
            System.out.printf("1");
            int data = 5 / 0;
        }
        catch(ArithmeticException e)
        {
            Throwable obj = new Throwable("Sample");
            try
            {
                throw obj;
            }
            catch (Throwable e1)
            {
                System.out.printf("8");
            }
        }
        finally
        {
            System.out.printf("3");
        }
        System.out.printf("4");
    }
}
```

☐ Compilation error☐ Runtime error☒ 1834☐ 134

✓



- ✓ public class Test { * 1/1
- ```
public static void main(String[] args) {
 int count = 1;
 while (count <= 15) {
 System.out.println(count % 2 == 1 ? "***" : "+++++");
 ++count;
 }
}
```
- ☐ 15 times \*\*\*
- ☐ 15 times +++++
- ☒ 8 times \*\*\* and 7 times +++++ ✓
- ☐ Both will print only once

- ✓ Which of these inheritances is shown in case we inherit some base class \*1/1  
from another class, then one of the classes derives it?
- ☐ Single
- ☐ Multiple
- ☒ Multi-level ✓
- ☐ Hierarchical



✓ What is the return type of the hashCode() method in the Object class? \* 1/1

- ☐ Object
- ☒ int
- ☐ long
- ☐ void



✗ Which of the following statements are incorrect? \* 0/1

- ☐ Variables declared as final occupy memory
- ☒ final variable must be initialized at the time of declaration
- ☐ Arrays in java are implemented as an object
- ☐ All arrays contain an attribute-length which contains the number of elements stored in the array



Correct answer

- ☒ Variables declared as final occupy memory

✓ In relation to abstraction, what does an abstract method represent? \* 1/1

- ☐ A method with a detailed implementation
- ☐ A method that is complete and cannot be overridden
- ☒ A method with no body, meant to be overridden in derived classes
- ☐ A method that cannot be used in an interface



✗ public class A \*

```
{
 public void toString()
 {
 System.out.println("Obj");
 }
 public static void main(String[] args)
 {
 A a = new A();
 a.toString();
 }
}
```

0/1

- ☐ Compilation Error
- ☐ Main@2a139a55
- ☒ Obj
- ☐ Runtime Error

✗

Correct answer

- ☒ Compilation Error





- ✓ public class MyFirst { \* 1/1
- ```
    public static void main(String[] args) {  
        MyFirst obj = new MyFirst(n);  
    }  
    static int a = 10;  
    static int n;  
    int b = 5;  
    int c;  
    public MyFirst(int m) {  
        System.out.println(a + ", " + b + ", " + c + ", " + n + ", " + m);  
    }  
    {  
        b = 30;  
        n = 20;  
    }  
    static  
    {  
        a = 60;  
    }  
}
```
- ☐ 10, 5, 0, 20, 0
- ☐ 10, 30, 20
- ☐ 60, 5, 0, 20
- ☒ 60, 30, 0, 20, 0 ✓

- ✓ Which of the following is true about the anonymous inner class? * 1/1
- ☐ It has only methods
- ☐ Objects can't be created
- ☐ It has a fixed class name
- ☒ It has no class name ✓

✓ public class Main{

*

1/1

```
    static String name = "Ramesh";
```

```
    public Main(){  
        name = "Prabhas";  
    }
```

```
    public static void main(String[] args){  
        System.out.println("The name is " + name);  
    }  
}
```

- ☐ Prabhas
- ☒ The name is Ramesh
- ☐ No Output
- ☐ Run-time error

✓

✓ Aggregation is which of the following? *

1/1

- ☒ Expresses a part-of relationship and is a stronger form of an association relationship.
- ☐ Expresses a part-of relationship and is a weaker form of an association relationship.
- ☐ Expresses an is-a relationship and is a stronger form of an association relationship.
- ☐ Expresses an is-a relationship and is a weaker form of an association relationship.

✓



✓ An interface with no fields or methods is known as a _____. *

1/1

- ☐ Runnable Interface
- ☒ Marker Interface
- ☐ Abstract Interface
- ☐ CharSequence Interface



This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#).

Google Forms



