

1. What is Java?

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

2. Mention some features of Java?

Some of the features which play important role in the popularity of java are as follows:

- Simple: Java is easy to learn. Eventhough Java is based on C++ , it was developed by eliminating poor programming practices of C++.
- Object-Oriented: Java is a object oriented programming language. Everything in Java is an Object.

[Learn more here.](#)

- Portable: Java run time environment uses a bytecode verification process to make sure that code loaded over the network doesn't violate Java security constraints.
- Platform independent: Java is platform independent. Java is a write once, run anywhere language. Without any modifications, we can use a program in different platforms.
- Secured: Java is well known for its security. It delivers virus free systems.
- High Performance: Java enables high performance with the use of JIT (Just-In-Time) compilers
- Multithreaded: Java Multithreaded features allows us to write programs that can perform many tasks simulatenously. Multithreading concept of Java shares a common memory area. It doesn't occupy memory for each thread.

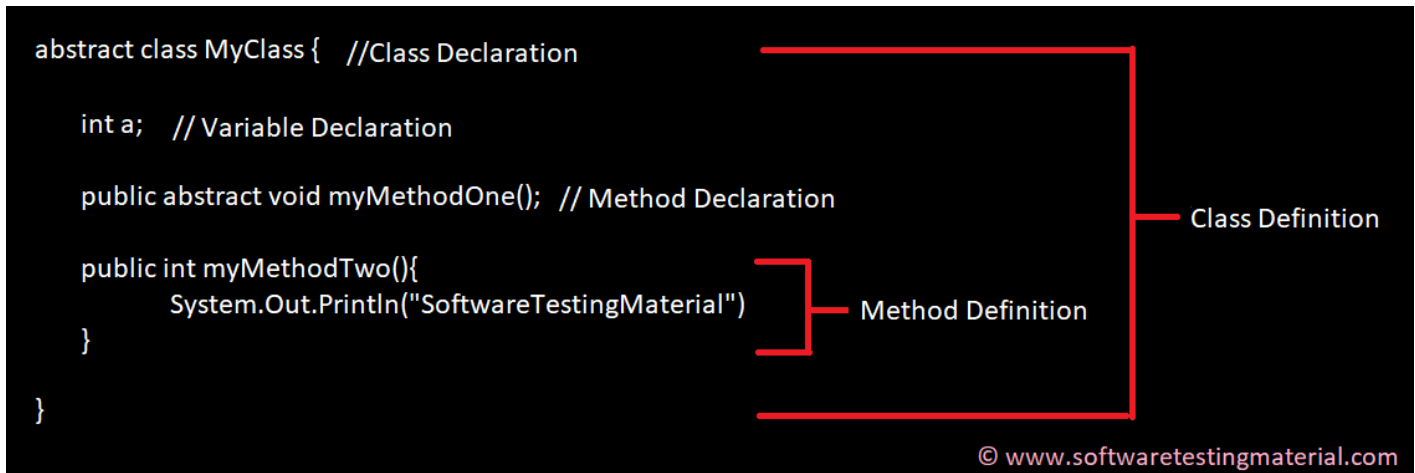
3. What is the difference between Declaration and Definition in Java?

Declaration: If you just declare a class or method/function or variable without mentioning anything about what that class or method/function or variable looks like is called as declaration in Java.

Definition: If you define how a class or method/function or variable is implemented then it is called definition in Java.

When we create an interface or abstract class, we simply declare a method/function but not define it.

For clear understanding, check the below image



4. What is an Object in Java?

An object is an instance of a class. Objects have state (variables) and behavior (methods).

Example: A dog is an object of Animal class. The dog has its states such as color, name, breed, and behaviors such as barking, eating, wagging her tail.

```

1 public class MyClass{ //Class name (MyClass) declaration
2     public static void main(String[] args){
3         MyClass obj = new MyClass(); //Object Creation
4     }
5 }

```

5. What is a Class in Java?

A class can be defined as a collection of objects. It is the blueprint or template that describes the state and behavior of an object.

```

1 public class MyClass{ //Class name (MyClass) declaration
2     int a = 9; // Variable declaration
3     int b = 99;
4     public void myMethod(){ //Method (myMethod) declaration
5         int sum=a+b;
6     }
7 }

```

6. What is Constructor in Java?

Constructor in Java is used in the creation of an Object that is an instance of a Class. Constructor name should be same as class name. It looks like a method but its not a method. It wont return any value. We have seen that methods may return a value. If there is no constructor in a class, then compiler automatically creates a default constructor.

7. What is Local Variable and Instance Variable?

Local Variable:

Local variable is a variable which we declare inside a Method. A method will often store its temporary state in local variables.

Instance Variable (Non-static):

Instance variable is a variable which is declared inside a Class but outside a Method. We don't declare this variable as Static because these variables are non-static variables.

Read more about [Variables in Java](#) here

8. What are the OOPs concepts?

OOPS Stands for Object Oriented Programming System. It includes Abstraction, Encapsulation, Inheritance, Polymorphism, Interface etc.,

Read more on [OOPs concept in Java](#)

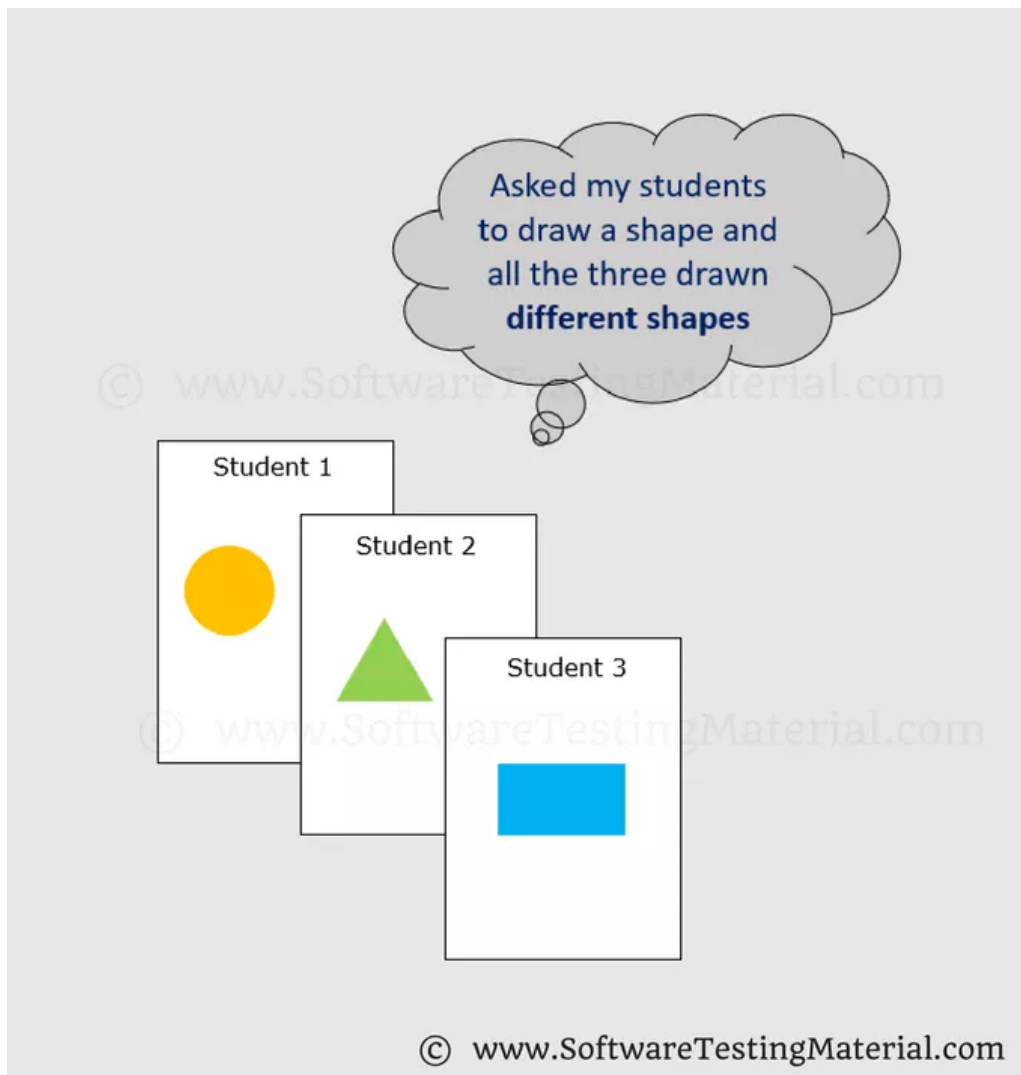
9. What is Inheritance in Java?

Inheritance is a process where one class inherits the properties of another class. [Read more here](#)



10. What is Polymorphism?

Polymorphism allows us to perform a task in multiple ways. Let's break the word Polymorphism and see it, 'Poly' means 'Many' and 'Morphos' means 'Shapes'. [Read more here](#)



Assume we have four students and we asked them to draw a shape. All the four may draw different shapes like Circle, Triangle, and Rectangle.

11. What are the types of Polymorphism?

There are two types of Polymorphism in Java

1. Compile time polymorphism (Static binding) – Method overloading
2. Runtime polymorphism (Dynamic binding) – Method overriding

We can perform polymorphism by 'Method Overloading' and 'Method Overriding'

12. What is Method Overloading?

A class having multiple methods with same name but different parameters is called Method Overloading

There are three ways to overload a method.

- Parameters with different data types
- Parameters with different sequence of a data types
- Different number of parameters

Read more on [Method Overloading in Java](#)

13. What is Method Overriding?

Declaring a method in child class which is already present in the parent class is called Method Overriding.

In simple words, overriding means to override the functionality of an existing method.

In this case, if we call the method with child class object, then the child class method is called. To call the parent class method we have to use **super** keyword.

Read more on [Method Overriding](#)

14. What is Abstraction in Java?

Abstraction is the methodology of hiding the implementation of internal details and showing the functionality to the users.



Example: Mobile Phone.

A layman who is using mobile phone doesn't know how it works internally but he can make phone calls.

15. What is Abstract Class in Java?

We can easily identify whether a class is an abstract class or not. A class which contains abstract keyword in its declaration then it is an Abstract Class.

Syntax:

```
1 abstract class <class-name>{}
```

Points to remember:

- Abstract classes may or may not include abstract methods
- If a class is declared abstract then it cannot be instantiated.
- If a class has abstract method then we have to declare the class as abstract class
- When an abstract class is subclassed, the subclass usually provides implementations for all of the abstract methods in its parent class. However, if it does not, then the subclass must also be declared abstract.

16. What is Abstract Method?

An abstract method is a method that is declared without an implementation (without braces, and followed by a semicolon), like this:

```
1 abstract void myMethod();
```

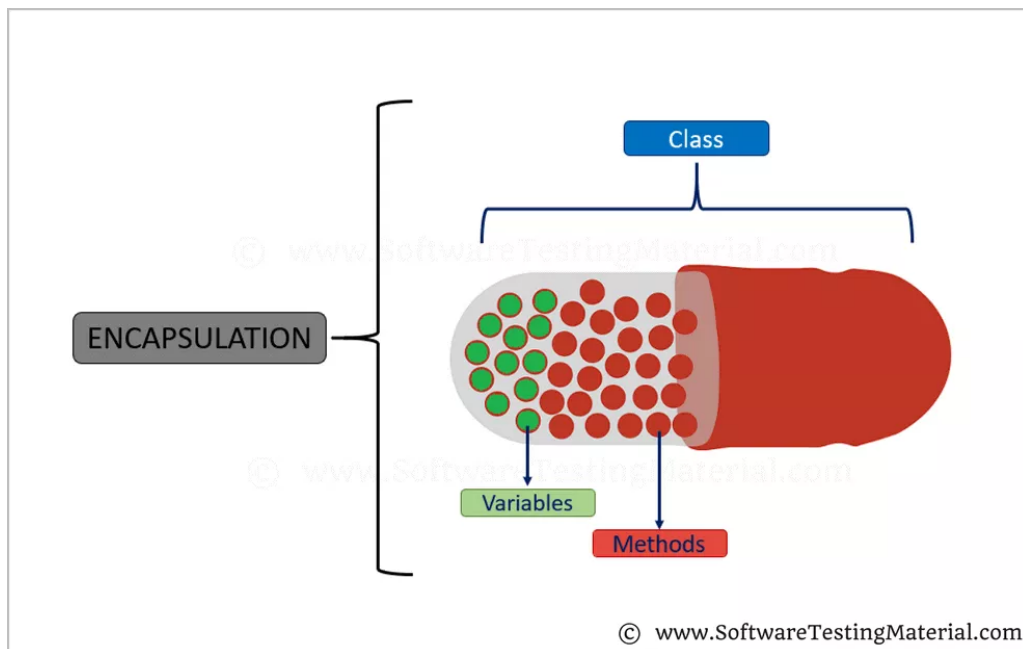
In order to use an abstract method, you need to override that method in sub class.

17. What is Interface in Java?

An interface in Java looks similar to a class but both the interface and class are two different concepts. An interface can have methods and variables just like the class but the methods declared in interface are by default abstract. We can achieve 100% abstraction and multiple inheritance in Java with Interface. Read more on [Interface in Java](#).

18. What is Encapsulation in Java?

Encapsulation is a mechanism of binding code and data together in a single unit. Let's take an example of Capsule. Different powdered or liquid medicines are encapsulated inside a capsule. Likewise in encapsulation, all the methods and variables are wrapped together in a single class. Read more on [Encapsulation in Java](#)



19. Write a program to print the pattern given below

```

1 1
2 1 2
3 1 2 3
4 1 2 3 4
5 1 2 3 4 5

```

Here is the program to print the pattern mentioned above

```

1 package softwareTestingMaterial;
2
3 public class NumberPattern {
4
5     public static void main(String[] args) {
6         for (int x = 1; x <= 5; x++) {
7             for (int y = 1; y <= x; y++) {
8                 System.out.print(y+" ");
9             }
10            System.out.println();
11        }
12    }
13 }

```

20. Write a program to print Fibonacci Series up to count 10.

```

1 package softwareTestingMaterial;
2
3 public class FibonacciSeries {
4
5     public static void main(String args[]) {
6         int a = 0, b = 1, c, i, count = 10;
7         // To print 0 and 1
8         System.out.print(a + " " + b);
9         // loop starts from 2. We have already printed 0 and 1 in the previous step
10        for (i = 2; i < count; i++) {
11            c = a + b;
12            System.out.print(" " + c);
13            a = b;
14            b = c;
15        }
16    }
17 }

```

21. How to reverse a String in Java?


```
1 package softwareTestingMaterial;
2
3 public class ReverseString {
4
5     public static void main(String[] args) {
6         // Using StringBuffer class
7         StringBuffer a = new StringBuffer("Software Testing Material");
8         // use reverse() method to reverse string
9         System.out.println(a.reverse());
10    }
11
12 }
```

Another method:

```
1 package softwareTestingMaterial;
2
3 public class ReverseString {
4
5     public static void main(String[] args) {
6         String input="Software Testing Material";
7         StringBuilder input1 = new StringBuilder();
8         input1.append(input);
9         input1=input1.reverse();
10        for (int i=0;i<input1.length();i++)
11            System.out.print(input1.charAt(i));
12    }
13
14 }
```

22. How To Find The Largest Value From The Given Array.

```
1 package softwareTestingMaterial;
2
3 public class LargestValue {
4
5     public static void main(String[] args){
6
7         int[] arr={28,3,15,9,17,4,23,2};
8
9         int val=arr[0];
10
11        for(int i=0; i<arr.length; i++){
12            if(arr[i] > val){
13                val=arr[i];
14            }
15        }
16        System.out.println("Largest value in the Given Array is "+ val);
17    }
18 }
```

23. How to display all the prime numbers between 1 and 100

The number which is only divisible by 1 and itself is known as a prime number. For example 2, 3, 5, 7, 11... are prime numbers.

```
1 package softwareTestingMaterial;
2
3 public class PrimeNumbersOneToHundred {
4     public static void main (String[] args){
5         int i =0;
6         int num =0;
7         String primeNumbers = "";
8
9         for (i = 1; i <= 100; i++){
10             int counter=0;
11             for(num =i; num>=1; num--){
12                 if(i%num==0){
```



```

13         counter = counter + 1;
14     }
15 }
16 if (counter ==2){
17     primeNumbers = primeNumbers + i + " ";
18 }
19 }
20 System.out.println("Prime numbers from 1 to 100 are :");
21 System.out.println(primeNumbers);
22 }
23 }

```

24. How to display all the prime numbers between 1 and n (n is the number, get the input from user)

```

1 package softwareTestingMaterial;
2
3 import java.util.Scanner;
4
5 public class PrimeNumbersOneToN {
6     public static void main (String[] args){
7         Scanner scanner = new Scanner(System.in);
8         int i =0;
9         int num =0;
10        String primeNumbers = "";
11        System.out.println("Enter the value of n :");
12        int n = scanner.nextInt();
13        scanner.close();
14        for (i = 1; i <= n; i++)
15        {
16            int counter=0;
17            for(num =i; num>=1; num--)
18            {
19                if(i%num==0)
20                {
21                    counter = counter + 1;
22                }
23            }
24            if (counter ==2)
25            {
26                primeNumbers = primeNumbers + i + " ";
27            }
28        }
29        System.out.println("Prime numbers from 1 to n are :");
30        System.out.println(primeNumbers);
31    }
32 }

```

25. How to find the given number is a prime number or not by getting input from the user

```

1 package softwareTestingMaterial;
2
3 import java.util.Scanner;
4
5 public class PrimeNumberVerification {
6     public static void main(String args[])
7     {
8         int i, j, flag = 0;
9         System.out.print("Enter any number which you want to verify whether it is a prime number or not");
10        Scanner s = new Scanner(System.in);
11        j = s.nextInt();
12        for( i = 2; i < j; i++){
13            if(j % i == 0){
14                flag = 0;
15                break;
16            }
17            else
18            {
19                flag = 1;

```

```
20         }
21     }
22     if(flag == 1){
23         System.out.println(j+" is a prime number.");
24     }
25     else{
26         System.out.println(+j+" is not a prime number.");
27     }
28 }
29 }
```

26. Write a program to print Fibonacci Series

Method 1:

```
1 package softwareTestingMaterial;
2
3 public class FibonacciSeries {
4
5     public static void main(String args[]) {
6         int a = 0, b = 1, c, i, count = 10;
7         // To print 0 and 1
8         System.out.print(a + " " + b);
9         // loop starts from 2. We have already printed 0 and 1 in the previous step
10        for (i = 2; i < count; i++) {
11            c = a + b;
12            System.out.print(" " + c);
13            a = b;
14            b = c;
15        }
16    }
17 }
```

Method 2:

```
1 package softwareTestingMaterial;
2
3 import java.util.Scanner;
4
5 public class FibonacciSeriesOne {
6     public static void main(String[] args){
7         System.out.println("Enter Iteration to print Fibonacci Series");
8         FibonacciCheck.checkFibonacci(new Scanner(System.in).nextInt());
9     }
10 }
11
12 class FibonacciCheck {
13     public static void checkFibonacci(int number) {
14         int first=0,second=1;
15         int third=0;
16         int i=1;
17         System.out.print("Fibonacci Series upto: "+number+" is ");
18         System.out.print(first+","+second+",");
19         while(i<=number){
20             third=first+second;
21             System.out.print(third+",");
22             first=second;
23             second=third;
24             ++i;
25         }
26     }
27 }
```

27. Difference between Array and ArrayList?

ARRAY

ARRAYLIST

ARRAY**ARRAYLIST**

Array is static

ArrayList is dynamic

Size of the array should be given at the time of array declaration. We cannot change the size of array after creating it

Size of the array may not be required. It changes the size dynamically. Capacity of ArrayList increases automatically whenever we add elements to an ArrayList

Array can contain both primitive data types as well as objects

ArrayList cannot contain primitive data types. It contains only objects

Arrays are multidimensional

ArrayList is always single dimension

28. Difference between ArrayList and HashSet in Java?**ARRAYLIST****HASHSET**

ArrayList implements List interface

HashSet implements Set interface

ArrayList allows duplicates

HashSet doesn't allow duplicates

ArrayList is an ordered collection and maintains insertion order of elements

HashSet is an unordered collection and doesn't maintain insertion order

ArrayList is backed by an Array

HashSet is backed by an HashMap instance

ArrayList is an index based

HashSet is object based

In ArrayList, we can retrieve object by calling get() method or remove object by calling remove() method

In HashSet, we can't achieve get() method

Learn more on [Array](#) and [ArrayList](#) with sample programs