1)What is constructor?

A constructor is basically a method that is automatically called when an object(instance) is created of that class. It is used to initialize an object's data members.

- Parameter Constructor
- Default Constructor

Example:

```
public class main(){

class main(){

class 5;

}

public static void main(String[] args){

main myObj=new main();

System.out.println(myObj.x);

}

}
```

Types Of Constructor

1. Default Constructor
2. Parameterized Constructor

------>Parameter Constructor:

Parameterized Constructor is used when it accepts a specific number of parameters. To initialize data members of a class with distinct values.

Example:

```
public class Yamini

{

string studentName;
int studentAge;
Yamini(String name, int age)
{
studentName = name;
studentAge = age;
}
void display(){
System.out.println(studentName+ " "+studentAge);
}
public static void main(String args[])
{
Yamini myObj = new Yamini("Manan",19);
myObj.display();
}
}
```

Default Constructor – Constructor that accepts no parameter is called Default Constructor. It is not necessary to have a constructor block in your class definition. If you don't explicitly write a constructor, the compiler automatically inserts one for you.

Example:

```
public class Yamini{
Yamini()
{
System.out.println("My name is Yamini");}
public static void main(String args[]){
Yamini obj = new Yamini();
}
}
```

2) Local Variables

A variable defined within a block or method or constructor is called a local variable.

- These variables are created when the block is entered or the function is called and destroyed after exiting from the block or when the call returns from the function.
- The scope of these variables exists only within the block in which the variable is declared. i.e. we can access these variables only within that block.
- Initialization of the local variable is mandatory before using it in the defined scope.

3) Instance Variables

Instance variables are non-static variables and are declared in a class outside any method, constructor or block.

- As instance variables are declared in a class, these variables are created when an object of the class is created and destroyed when the object is destroyed.
- Unlike local variables, we may use access specifiers for instance variables. If we do not specify any access specifier then the default access specifier will be used.
- Initialization of Instance Variable is not Mandatory. Its default value is 0
- Instance Variable can be accessed only by creating objects.

4) Static Variables

Static variables are also known as Class variables.

- These variables are declared similarly as instance variables, the difference is that static variables are declared using the static keyword within a class outside any method constructor or block.
- Unlike instance variables, we can only have one copy of a static variable per class irrespective of how many objects we create.
- Static variables are created at the start of program execution and destroyed automatically when execution ends.
- Initialization of Static Variable is not Mandatory. Its default value is 0
- If we access the static variable like the Instance variable the compiler will show the warning message and it won't halt the

program. The compiler will replace the object name with the class name automatically.

- If we access the static variable without the class name, the compiler will automatically append the class name.

5)Garbage collector

Java garbage collection is the process by which Java programs perform automatic memory management. Java programs compile to bytecode that can be run on a Java Virtual Machine, or JVM for short. The garbage collector finds these unused objects and deletes them to free up memory.

6)Object Count

In Java, when we create an object of the class, the constructor of the class is always called, by default. In order to count the number of objects, we need to add a count variable in the constructor and increments its value by 1 for each invocation. Remember that the variable count must a class-level variable.