1)What is constructor?

A constructor is basically a method that is automatically calledwhenanobject(instance) is created of that class. It is used to initialize an object'sdatamembers.

- 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 numberofparameters. To initialize data members of a class with distinct values.

Example:

```
public class Laxmi

{
```

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

Default Constructor – Constructor that accepts no parameter is calledDefaultConstructor. 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 Balu{
Yamini()
{
System.out.println("My name is Balu");}
public static void main(String args[]){
Balu obj = new Balu();
}
}
```

2) Local Variables

A variable defined within a block or method or constructor is calledalocal variable.

- These variables are created when the block is entered or thefunction is called and destroyed after exiting fromthe block or whenthe call returns from the function.
- The scope of these variables exists only within the block inwhichthe variable is

declared. i.e. we can access these variables onlywithin that block.

· Initialization of the local variable is mandatory before using it inthedefined scope.

3) Instance Variables

Instance variables are non-static variables and are declared inaclassoutside any method, constructor or block.

· As instance variables are declared in a class, these variables arecreated when an object of the class is created and destroyedwhenthe object is destroyed.

· Unlike local variables, we may use access specifiers for instancevariables. If we do not specify any access specifier then thedefaultaccess specifier will be used.

· Initialization of Instance Variable is not Mandatory. Its default valueis 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, thedifference 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 staticvariable per class irrespective of how many objects we create. · Static variables are created at the start of programexecutionanddestroyed automatically when execution ends.

· Initialization of Static Variable is not Mandatory. Its default valueis0

· If we access the static variable like the Instance variable thecompiler will show the warning message and it won't halt the program. The compiler will replace the object name with theclassname automatically.

· If we access the static variable without the class name, thecompilerwill automatically append the class name.

5)Garbage collector

Java garbage collection is the process by which Java programs performautomatic memory management. Java programs compile to bytecodethat canbe run on a Java Virtual Machine, or JVM for short. The garbage collector findsthese 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 theclass is

always called, by default. In order to count the number of objects, weneed to add a count variable in the constructor and increments its valueby1 for each invocation. Remember that the variable count must a class-level variable.

Example:

```
public class countObj {
static countObj=0;
{
countObj++
}
public static void main(String args[]){
      countObj object1=new countObj();
      countObj object2=new countObj();
      countObj object3=new countObj();
      countObj object4=new countObj();
      countObj object5=new countObj();
      countObj object6=new countObj();
      System.out.println("number of objects is created"+countObj); }
}
```