10 Feb'23 Constructor Assignment - 17

1. What is a Constructor?

Ans: 1. Object creation is not enough, compulsorily we should perform **initialization** then only the **object** is in a position to provide the response properly.

2. Whenever we are creating an object some piece of the code will be executed automatically to perform **initialization of an object**. This piece of code is nothing but a **constructor**.

3. Main objective of the **constructor** is nothing but **initialisation of Object**.

Rules of writing a constructor:

1. Name of the constructor and name of the class must be the same.

2. Return type concept not applicable for constructor, even if we provide it won't result in compile time errors, if we do so then the Java language will treat this as "normal method".

Eg

```
class Test{
void Test(){
System.out.println("Hello");// It is not a constructor,it is a method.
}
}
```

- It is not a good practice to take the method name same as that of the classname.
- The modifiers applicable for constructors are private, public, protected, default.
- The other modifiers if we use, it would result in compile time error.

```
class Test{
static Test(){
}
}
```

10 Feb'23 Constructor Assignment - 17

2. What is Constructor Chaining?
Ans: A class can contain more than one constructor and all these constructors have the same name; they differ only in the type of argument, hence these constructors are considered as **"Overloaded constructor"**.

3. Can we call a subclass constructor from a superclass constructor?
Ans: A **subclass** can call a **constructor** defined by its **superclass** by use of the following form of **super**: **super(parameter-list);** Here, parameter-list specifies any parameters needed by the **constructor** in the **superclass**.

The superclass constructor can be called from the first line of a subclass constructor by using the keyword super and passing appropriate parameters to set the private instance variables of the superclass.

4. What happens if you keep a return type for a constructor?
Ans: Constructor does not have any return type in Java. Constructor looks like a method but it is not. It does not have a return type and its name is the same as the class name.

Since the constructor can only return the object to class, it's implicitly done by java runtime and we are not supposed to add a return type to it.

If we add a return type to a constructor, then it will become a method of the class. This is the way java runtime distinguishes between a normal method and a constructor.

5. What is No-arg constructor?
Ans: Here, the constructor does not accept any parameters. Hence, it is known as a no-arg constructor.

For example, the no-arg constructor is invoked when a variable is declared like this: **Complex c = new Complex ()**; If there are no constructors defined in a Java class, the Java compiler provides a default no-arg constructor .
The default no-arg constructor does nothing.

10 Feb'23 Constructor Assignment - 17

6. How is a No-argument constructor different from the default constructor?
Ans: It is created to assign the default values to the instance variables of the class when an object is created. Default constructors are sometimes called no-arg constructors since they both work the same.
But no-arg constructor is created by the user while default constructor can only be created by the compiler.

Default Constructor in java:

- When we write a class without any constructor then at compilation time java compiler creates a default constructor in our class.
- The accessibility modifier of the default constructor is the same as the accessibility modifier of class.
- The allowed accessibility modifiers are public and default.
- Default constructor added by java compiler this constructor does not have anything except **super();** call.
- If our class have any constructor then java compiler does not create default constructor

Code Example:

```
package constructor;
public class A {


}
```

```
package constructor;
public class A {

    A(){

        super();
```

10 Feb'23 Constructor Assignment - 17

```
    }

}
```

No-argument Constructor in java:

- As a developer we can create our own constructor with no arguments is known as no-argument constructor.
- It can have all four accessibility modifiers as it is defined by the developer.
- So allowed accessibility modifiers are public, private, protected and default
- It can have logic including super calls.
- The common point between default and no-argument constructor Both do not have any arguments.
- And one more point we need to remember that in no-argument constructor also by default the first statement will be a super() call which is added by the java compiler if it does not have.

Example code:

```
package constructor;
public class A {

    A(){

  super();
  System.out.println("no -argument constructor");

    }

}
```

10 Feb'23 Constructor Assignment - 17

7. When do we need constructor Overloading?

Ans: Constructor in java is used to create the instance of the class. Constructors are almost similar to methods except for two things - its name is the same as the class name and it has no return type.
Sometimes constructors are also referred to as special methods to initialize an object.

As construction overloading enables the creation of the object of a specific class in several ways, it is most commonly used in Java programs based on the requirement of the programmer.
With the use of constructor overloading, objects can be initialized with different data types.

8. What is Default constructor Explain with an Example

Ans:

● For every java class constructor concept is applicable.
● If we don't write any constructor, then the compiler will generate a default constructor.
● If we write at least one constructor then the compiler won't generate any default
constructor, so we say every java class will have a compiler generated default
constructor or programmer written constructor but not both simultaneously.