

**Q1. What is a Constructor?**

Ans. 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.

>Main Objective of constructor is to initialize the object.

**Q2. What is Constructor Chaining?**

Ans. When a class contains more than one constructor and all these constructors have the same name they differ only in the type of argument, hence these constructor are considered as "Constructor Chaining".

**Q3. Can we call a subclass constructor from a superclass constructor?**

Ans. No, we can not call a subclass constructor from a superclass constructor because the constructor never participates in inheritance.

**Q4. What happens if you keep a return type for a constructor?**

Ans. If we keep a return type for a constructor it will not show us a compile time errors, java language will treat this as "normal method".

**Q5. What is the No-argument constructor?**

Ans. A constructor which does not have any parameter type is called No-argument constructor.

Example:

```
Class Super1
{
    public Super1() //No argument constructor
    {
        System.out.println("Super class Constructor");
    }
}
```

**Q6. How is a No-arguments constructor different from the default Constructor?**

Ans. No-argument constructor is a constructor through which a programmer display some message during object initialization where as Default Constructor is generated by a compiler by default when we do not write any constructor inside a class

**Q7. Why do we need a constructor overloading?**

Ans. When a class contains more than one constructor and all these constructors have the same name they differ only in the type of argument, hence these constructor are considered as "Constructor Overloading"

> Through constructor overloading we can pass multiple parameters to the object during its initialization.

>We can perform different task with the objects.

**Example:**

```

class Test {
    Test(double d) {
        System.out.println("Double argument constructor");
    }
    Test(int i) {
        System.out.println("int argument constructor");
    }
    Test() {
        System.out.println("No argument constructor");
    }
}
public class Mainclass{
    public static void main(String[] args) {
        Test t1 = new Test();//no argument constructor
        Test t1 = new Test(12);//int argument constructor
        Test t1 = new Test(10.5);//double argument constructor
    }
}

```

#### **Q8. What is the Default Constructor Explain with an Example?**

Ans. Default Constructor is a constructor which is generated by the compiler itself. The access modifier of the default constructor is the same as the class modifier. [applicable for public and default].

>Default constructor contains one line, super(). It is a call to super class constructor

Example:

```

Class Super1
{
    public Super1()//Default constructor
    {
        super();
    }
}

```