Constructor

Constructor is a function same name as class name but without return type.

Syntax:

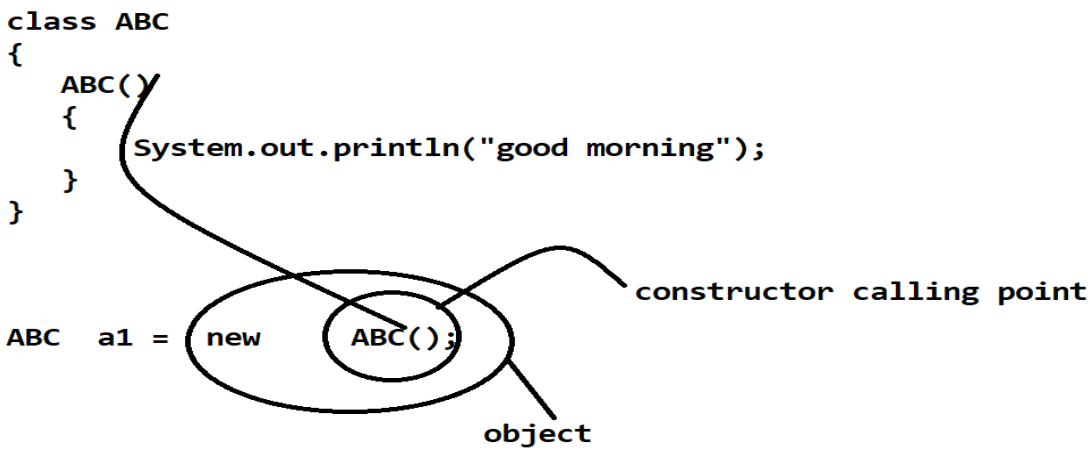
```
class classname
{
   classname()
   {
   }
}
```

Q. why use constructor or what is the benefit of constructor?

The major goal of constructor is

1) Call the function automatically when class object get created.

2) To initialize the memory of object means when we call the constructor then JVM allocate memory for object.



Source Code

```
package org.techhub;

class ABC
{
	ABC()
	{
		System.out.println("I am constructor");
	}
}
public class ConstructorApplication {

	 public static void main(String x[])
	 {
		ABC a1 = new ABC();
	 }
}
```

Types of constructor

1) Default constructor : default constructor means if we not pass parameter to the constructor called as default constructor.

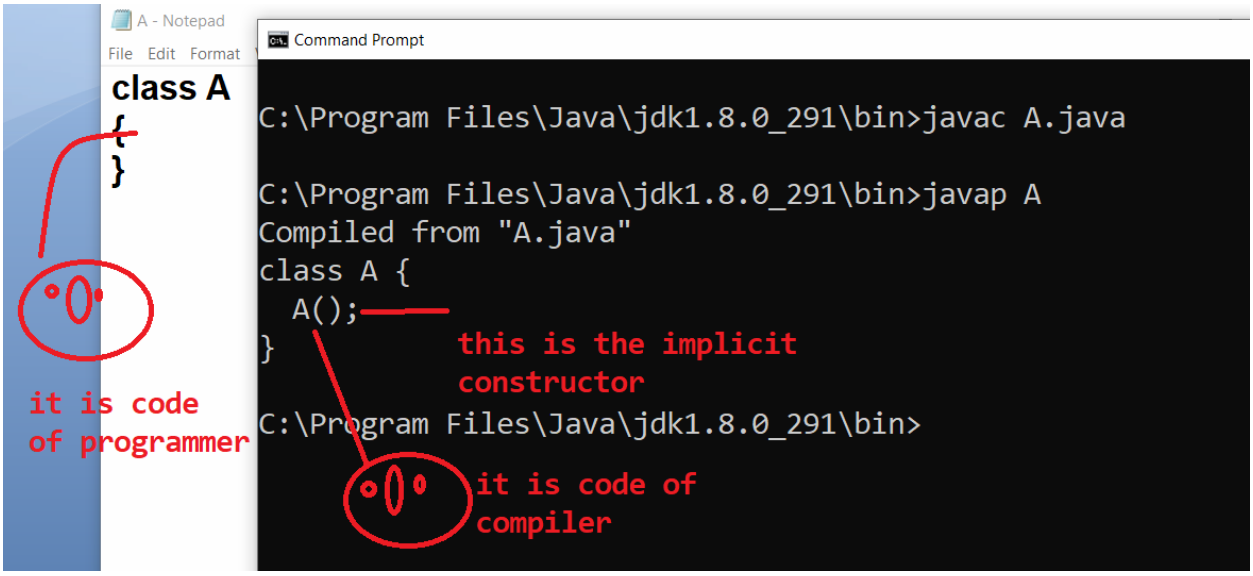
```
class ABC
{
	ABC()
	{
		System.out.println("I am constructor");
	}
}
public class ConstructorApplication {
```

```
        public static void main(String x[])
        {
                ABC a1 = new ABC();
        }
}
```

Note: if we not declare the constructor within class then java add the by default constructor in class called as implicit constructor.

Can we see the implicit constructor in java?

Yes we can see the implicit constructor in java



2) Argument constructor: Augmented constructor or Parameterized constructor if we pass parameter to constructor called as parameterized constructor. When we pass the parameter to constructor then we need to pass parameter from where object get created.

Following Example shows the Parameterized constructor

```
class Add
{  int a,b;
    Add(int a,int b)
    {
           this.a=a;
           this.b=b;
    }
   void showAdd()
   {
    System.out.printf("Addition is %d\n",a+b);
   }
}
public class AddApplication
{
    public static void main(String x[])
    {
             Add  ad = new Add(100,200);
             ad.show();
    }
}
```

100,200

3) Overloaded constructor:

Before constructor overloading we need to know what is the overloading.

Overloading is part of compile time polymorphism.

What is the meaning of Polymorphism?

Polymorphism means if we use the same thing for different purpose called as polymorphism
Poly means many and morphs means form called as polymorphism

Example of Polymorphism is

Suppose consider Person is best example of Polymorphism
He can change its behavior according its requirement
Suppose person is employee of company Then his behavior like as employee and same person is parent then He can change its behavior as parent if same person is husband then He can change its behavious like husband.

There are two types of polymorphism

a) Compile time polymorphism: compile time polymorphism method calling is decide at the time of program compilation called as compile time polymorphism.

There are two types of compile time polymorphism in java

i) Function overloading: function overloading means if we use the same function name with different parameter with different data type with different sequence called as function overloading.

Example

```
void add(int x,int y)
{
}
void add(float x,float y)
{
}
void add(int x,float y)
{

}
void add(float x,int y)
{

}
```

If we want to work with function overloading we have the some important points.

---

a) Function name should be same

b) Parameter list or parameter type must be different or minimum sequence should be different

c) In function overloading return type is not consider

Means there is no compulsion return must be same for all function you can give the different return type to the function

d) Which function get executed is depend on how much parameter pass in it and its sequence.

Can you give real time example where we can use the function overloading in project?

Suppose consider we have training center admission module implementation and training center takes two types of admission

A) Fresher admission

B) Working Professional admission

In this scenario we design two functions with same name

Void admission()

First function for fresher like as

_____

```
void admission(String name,String email,String collegeName,int
passyear)
{
}
```

Second function for Working Professional

```
void admission(int expinyear,String name,String email,String
contact,String compName,int currentCTC,int expectedPackage)
{
}
```

Following example demonstrate the function overloading

```
class Square
{
  void calSquare(int x)
   {
    System.out.printf("Square of integer is %d\n",x*x);
   }
   void calSquare(float x)
   {
    System.out.printf("Square of float is %f\n",x*x);
   }
}
public class SquareApplication
{
    public static void main(String x[])
    {
          Square s1 = new Square();
           s1.calSquare(5); //call integer function here
           s1.calSquare(5.4f); //call floating function
    }
}
```

ii) Constructor overloading.

b) Run time polymorphism

4) this() constructor