Agenda

Constructor
Types of Constructors
Constructor vs Method in Java
This keyword

Constructor

In Java, a constructor is a block of codes similar to the method.

It is called when an instance of the class/object is created, and memory is allocated for the object.

It is a special type of method which is used to initialize the object.

NOTE: **If there is no constructor in a class, compiler automatically creates a default constructor.**

Constructor

Object creation syntax:

Class-name reference-variable = new classname();

Test t =new Test();

Test ---> class Name
t ---> Reference variables
= ---> assignment operator
new ---> keyword used to create object
Test() ---> constructor ;
; ---> statement terminator

Constructor

When we create instance (Object) of a class using new keyword, a constructor for that class is called.

- new keyword is used to create object in java.

- When we create object by using new operator after new keyword that part is constructor and it is executing a constructor.

Rules to declare constructor

Rules to declare constructor:

1. Constructor not allowed any return type even void.

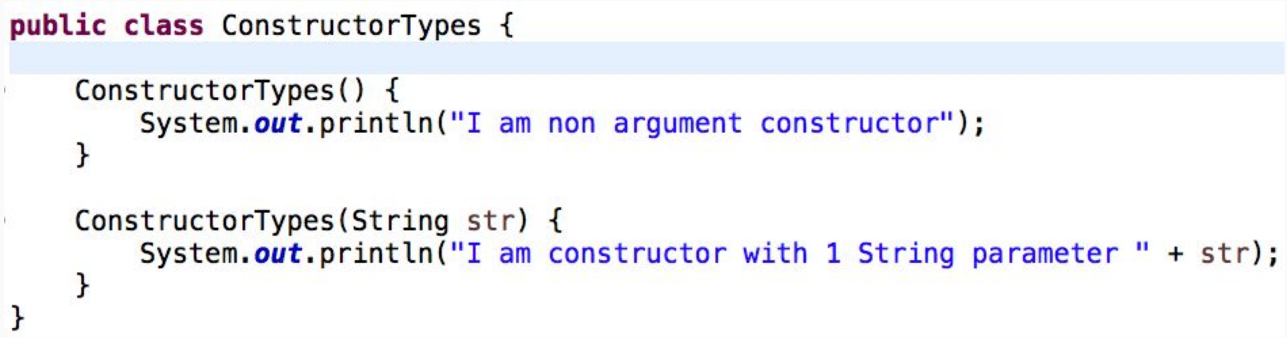
2. Constructor name class name must be same.

Type of constructors

There are two types of constructors:

1. Non argument ()

2. Parameterized



```
public class ConstructorTypes {

    ConstructorTypes() {
        System.out.println("I am non argument constructor");
    }

    ConstructorTypes(String str) {
        System.out.println("I am constructor with 1 String parameter " + str);
    }
}
```

Non Argument Constructor

- Inside the class if we are not declaring any constructor then compiler generates zero argument constructors with empty implementation at the time of compilation is called default constructor.

- The compiler generated constructor is called default constructor.

- Inside the class default constructor is invisible mode.

- The purpose of default constructor is to provide the default values to the object like 0, null etc. depending on the type.

Parameterized Constructor

- If any constructor contains list of variable in its signature is known as parameterized constructor. A parameterized constructor is one which takes some parameters.
- Parameterized constructor is used to provide different values to the distinct objects.

Syntax:

```
class ClassName {
     //parameterized constructor
     ClassName(list of parameters) {
      }
}
```

Syntax to call parameterized constructor

```
ClassName  objref=new ClassName(value1,
value2,.....);
             OR
new ClassName(value1, value2,.....);
```

Parameterized Constructor - Important points

- Whenever we create an object using parameterized constructor, it must be define parameterized constructor otherwise we will get compile time error.

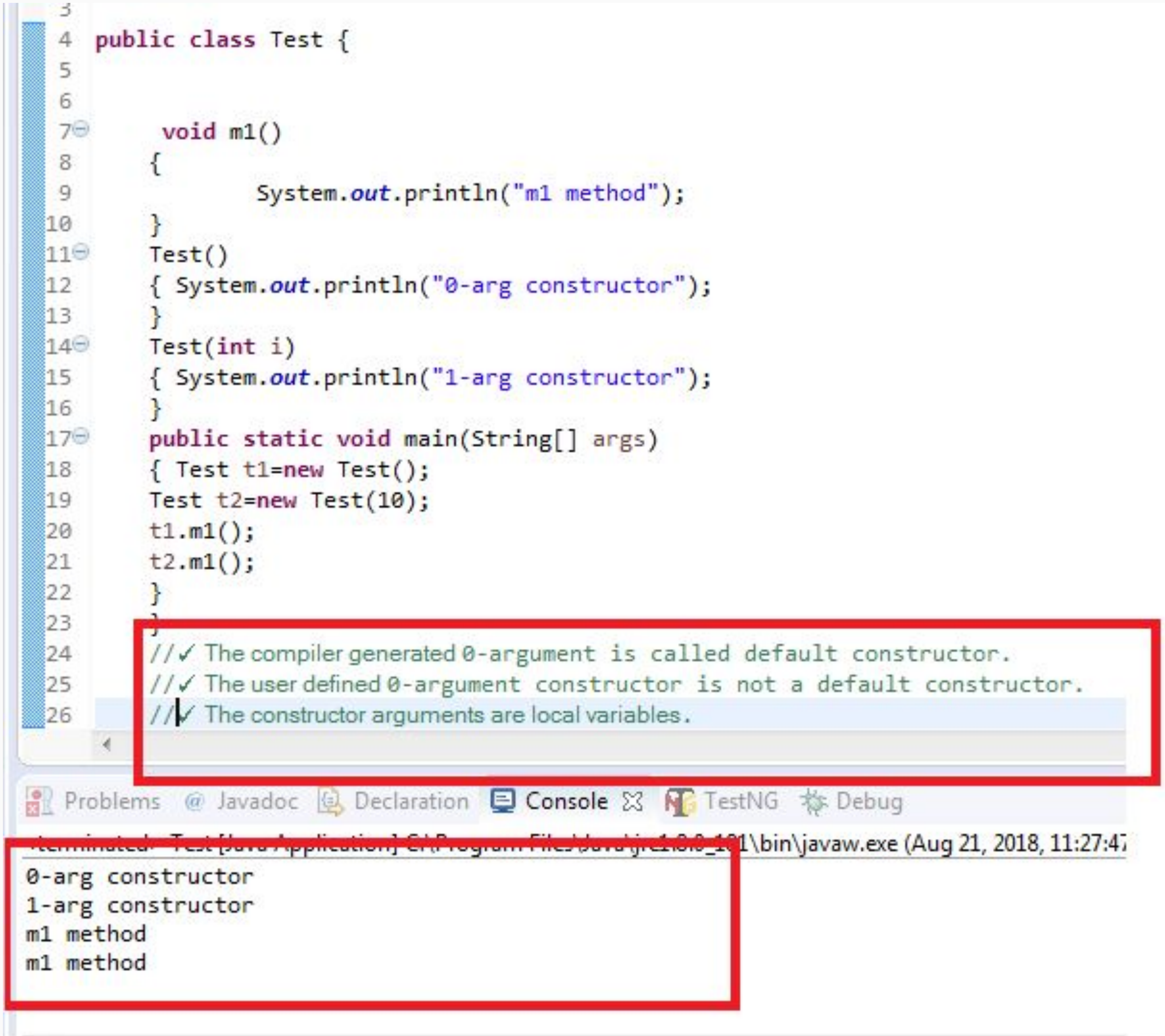
- Whenever we define the objects with respect to both parameterized constructor and default constructor, It must be define both the constructors.

- In any class maximum one default constructor but 'n' number of parameterized constructors.

Example of Parameterized Constructor

```
class Test {
   int a, b;
   Test(int n1, int n2) {
          SOP("I am from Parameterized Constructor...");
          a=n1;
          b=n2;
          SOP("Value of a = "+a);
          SOP("Value of b = "+b);
     }
};
class TestDemo1 {
   public static void main(String k []) {
   Test t1=new Test(10, 20);
   }
};
```

Sample Program

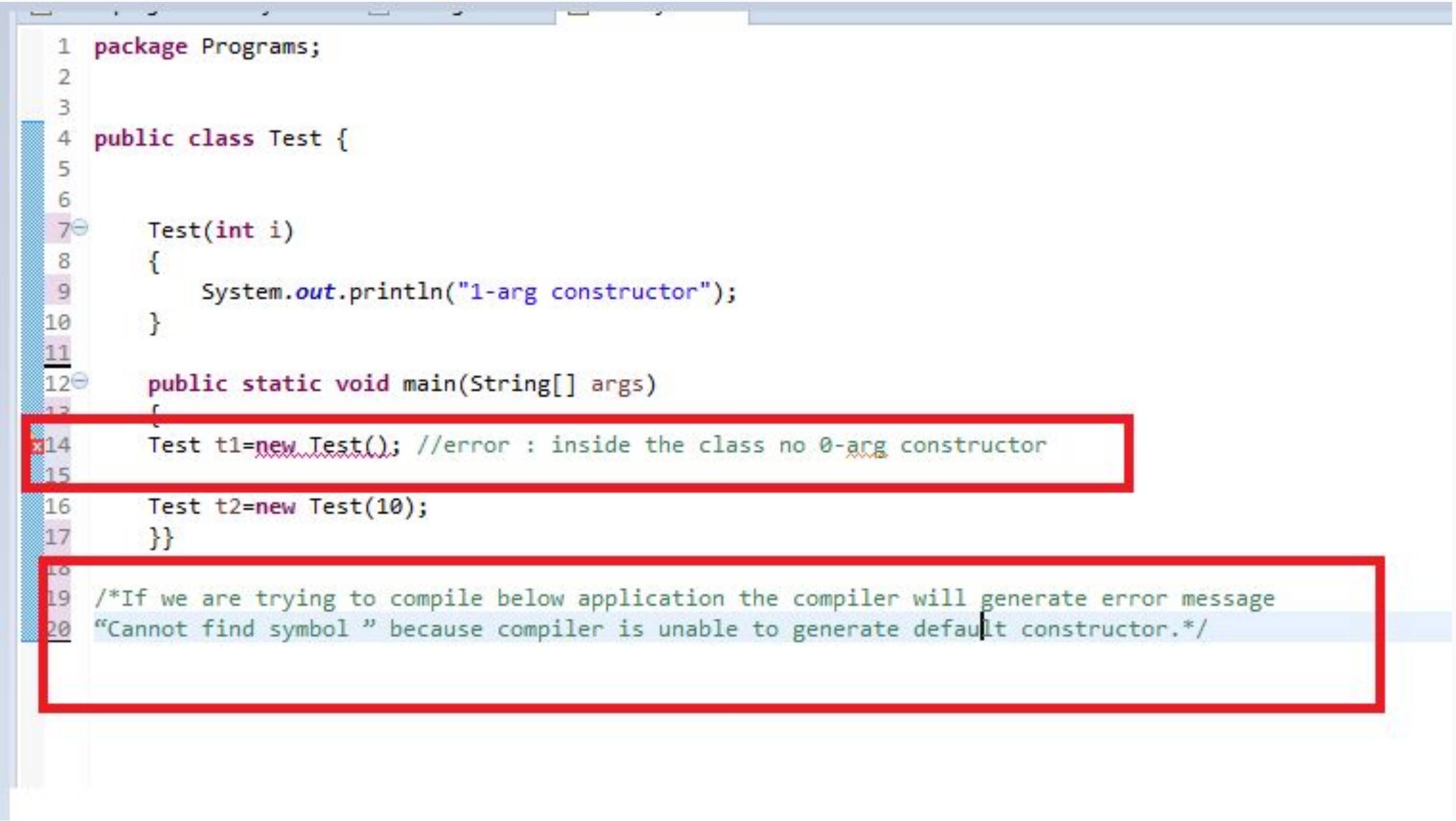


```
public class Test {


     void m1()
    {
            System.out.println("m1 method");
    }
    Test()
    { System.out.println("0-arg constructor");
    }
    Test(int i)
    { System.out.println("1-arg constructor");
    }
    public static void main(String[] args)
    { Test t1=new Test();
    Test t2=new Test(10);
    t1.m1();
    t2.m1();
    }
    }
    //✓ The compiler generated 0-argument is called default constructor.
    //✓ The user defined 0-argument constructor is not a default constructor.
    //✓ The constructor arguments are local variables.
```

```
<terminated> Test [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (Aug 21, 2018, 11:27:47
0-arg constructor
1-arg constructor
m1 method
m1 method
```

Sample Program



```
package Programs;


public class Test {


    Test(int i)
    {
        System.out.println("1-arg constructor");
    }

    public static void main(String[] args)
    {
    Test t1=new Test(); //error : inside the class no 0-arg constructor

    Test t2=new Test(10);
    }}

/*If we are trying to compile below application the compiler will generate error message
“Cannot find symbol ” because compiler is unable to generate default constructor.*/
```

Rules or properties of a constructor

1. Constructor will be called automatically when the object is created.
2. Constructor should not return any value even void also. Because basic aim is to place the value in the object. (if we write the return type for the constructor then that constructor will be treated as ordinary method).
3. Constructor definitions should not be static. Because constructors will be called each and every time, whenever an object is creating.
4. Constructor should not be private provided an object of one class is created in another class (Constructor can be private provided an object of one class created in the same class).
5. Constructors will not be inherited from one class to another class (Because every class constructor is create for initializing its own data members).

Method vs Constructor

Method	Constructor
Method can be any user defined name	Constructor must be class name
Method should have return type	It should not have any return type (even void)
Method should be called explicitly either with object reference or class reference	It will be called automatically whenever object is created
Method is not provided by compiler in any case.	The java compiler provides a default constructor if we do not have any constructor.
Method can use non access modifiers such as static, final, abstract	Constructor cannot use non access modifiers

Method vs Constructor

Constructors are used to write the logics these logics are executed during object creation.

Method also used to write the logics these logics are executed when we call the method.

This Keyword

this is a keyword in Java that refers to the current object.

this is used to access current class instance members

Usage:

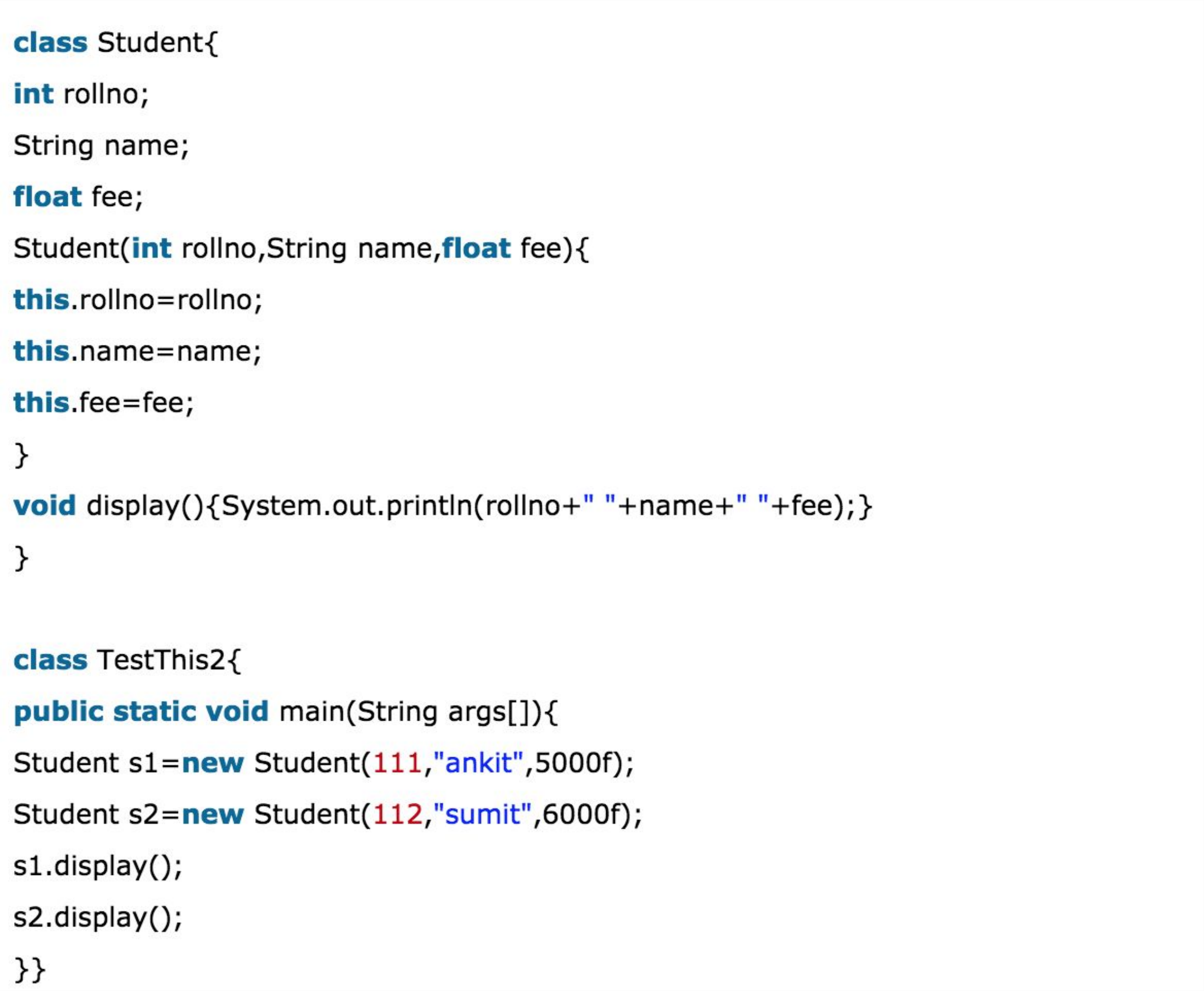
- With variables

  this can be used to refer current class instance variable.

- With methods
  this can be used to invoke current class method (implicitly).

- With constructors
  this() can be used to invoke the current class constructor.

This Keyword



```
class Student{
int rollno;
String name;
float fee;
Student(int rollno,String name,float fee){
this.rollno=rollno;
this.name=name;
this.fee=fee;
}
void display(){System.out.println(rollno+" "+name+" "+fee);}
}

class TestThis2{
public static void main(String args[]){
Student s1=new Student(111,"ankit",5000f);
Student s2=new Student(112,"sumit",6000f);
s1.display();
s2.display();
}}
```

This Keyword

```
class Message{
    Message(){
        System.out.println("hello a");
    }

    Message(int p){
        this();
        System.out.println(""p);
    }
}

class ThisKeyword{
    public static void main(String args[]){
    Message a=new Message(12);
    }
}
```

this vs this()

THIS	**THIS()**
Used with variables and methods	Used with constructors only
One of the uses is differentiate between local and instance variables in a method call	Used to call one constructor from another belonging to the same class