JAVA ASSIGNMENT :

K.SAI NAGA SRI

17B01A0477

ECE-B

1. WHAT IS A CONSTRUCTOR ?

A constructor initializes an object when it is created. It has the same name as its class and is syntactically similar to a method. However, constructors have no explicit return type.

Typically, you will use a constructor to give initial values to the instance variables defined by the class, or to perform any other start-up procedures required to create a fully formed object.

All classes have constructors, whether you define one or not, because Java automatically provides a default constructor that initializes all member variables to zero. However, once you define your own constructor, the default constructor is no longer used.

SYNTAX :

```
Class ClassName {
  ClassName() {
}}
```

Java allows two types of constructors namely –

- No argument Constructors
- Parameterized Constructors

No argument Constructors

As the name specifies the no argument constructors of Java does not accept any parameters instead, using these constructors the instance variables of a method will be initialized with fixed values for all objects.

Parameterized Constructors

Most often, you will need a constructor that accepts one or more parameters. Parameters are added to a constructor in the same way that they are added to a method, just declare them inside the parentheses after the constructor's name.

1 .EXAMPLE FOR “PARAMETERIZED CONSTRUCTORS” :

```java
//EXAMPLE FOR "PARAMETERIZED CONSTRUCTERS":
class SaiSri{
   double length;
   double breadth;
   double height;
   SaiSri(double l,double b,double h){
       length =l;
       breadth =b;
       height =h;
  }
   double volcomp(){
       double volume=length*breadth*height;
       return volume;
  }
}
public class SaiSric0{
   public static void main(String args[]){
    SaiSri p0= new SaiSri(10,20,30);
    SaiSri p1= new SaiSri(40,70,50);
    System.out.println("The volume of the room is"+""+p0.volcomp());
    System.out.println("The volume of the room is"+""+p1.volcomp());
  }
}
```

Execute Mode, Inputs & Arguments

Execute

Result

CPU Time: 0.09 sec(s), Memory: 27652 kilobyte(s)

```
The volume of the room is6000.0
The volume of the room is140000.0
```

File Explorer

2. DECLARATION OF A METHOD :

A method's declaration provides a lot of information about the method to the compiler, the runtime system and to other classes and objects. Besides the name of the method, the method declaration carries information such as the return type of the method, the number and type of the arguments required by the method, and what other classes and objects can call the method.

The only two required elements of a method declaration are the method name and the data type returned by the method.

SYNTAX FOR DEFINING A METHOD:

```
<access specifier><return type><method
 name>(parameter lit){
  //method definition
  }
```

2 . EXAMPLE FOR METHOD OVERLOADING :

For Multiple Files, Custom Library and File Read/

```java
class SaiSri {
    void test(String a, float b)
    {
        System.out.println("String float method");
    }
    void test(int a, double b)
    {
        System.out.println(" int double method");
    }
}
public class Saisri0{
    public static void main(String args[]){
        SaiSri  ob= new SaiSri();
        ob.test("pqr",90);
        ob.test(45,24.72d);
    }
}
```

Execute Mode, Inputs & Arguments

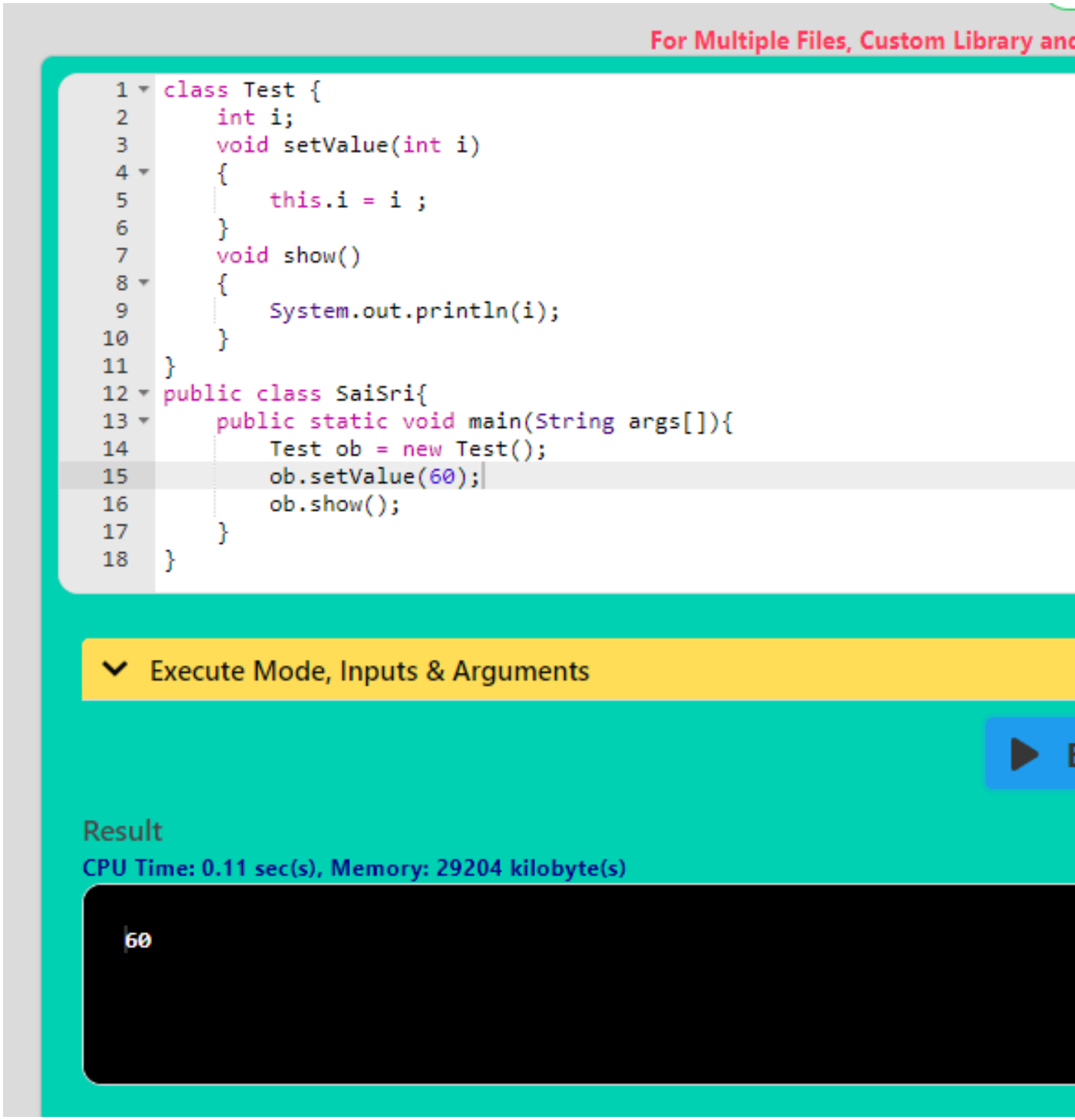
Execute

Result

CPU Time: 0.08 sec(s), Memory: 29128 kilobyte(s)

```
String float method
 int double method
```

3 . EXAMPLE PROGRAM FOR “this” KEYWORD :



4 . EXAMPLE PROGRAM FOR “STATIC” KEYWORD :

```
1  class SaiSri {
2      static void display() {
3          System.out.println("7");
4      }
5      void show() {
6          System.out.println("9");
7      }
8  }
9  public class SaiSri0{
10     public static void main(String args[]) {
11         SaiSri ob = new SaiSri();
12         ob.show();
13         ob.display();
14     }
15 }
```

Execute Mode, Inputs & Arguments

Execu

Result

CPU Time: 0.07 sec(s), Memory: 25440 kilobyte(s)

```
9
7
```

```java
class SaiSri {
    void test(String a, float b)
    {
        System.out.println("String float method");
    }
    void test(int a, double b)
    {
        System.out.println(" int double method");
    }
}
public class Saisri0{
    public static void main(String args[]){
        SaiSri  ob= new SaiSri();
        ob.test("pqr",90);
        ob.test(45,24.72d);
    }
}
```

Execute Mode, Inputs & Arguments

Execute

Result

CPU Time: 0.08 sec(s), Memory: 29128 kilobyte(s)

```
String float method
 int double method
```