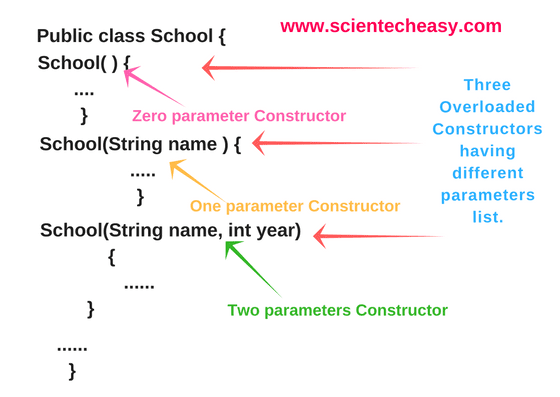
|  |  |
| --- | --- |
| **Name – Prerna Sunil Jadhav** | **SAP ID - 60004220127** |

**Experiment No – 07**

**AIM: To implement Constructors and overloading (CO2)**

**THEORY:**

**Constructor:**

* In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the memory.
* It is a special type of method which is used to initialize the object.
* Every time an object is created using the new() keyword, at least one constructor is called.
* It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default.

**Constructor Overloading:**

* In Java, we can overload constructors like methods.
* The constructor overloading can be defined as the concept of having more than one constructor with different parameters so that every constructor can perform a different task.

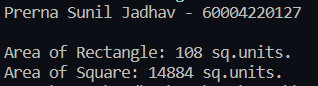
**CODE (i): WAP find area of square and rectangle using overloaded constructor**

****

**Text

Description automatically generated**

**OUTPUT:**

****

**CODE (ii): Create Rectangle and Cube class that encapsulates the properties of a rectangle and cube i.e. Rectangle has default and parameterized constructor and area() method. Cube has default and parameterized constructor and volume() method. They share no ancestor other than Object.**

**Implement a class Size with size() method. This method accepts a single reference argument z. If z refers to a Rectangle then size(z) returns its area and if z is a reference of Cube, then z returns its volume. If z refers to an object of any other class, then size(z) returns -1. Use main method in Size class to call size(z) method.**

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

**OUTPUT:**

**Text

Description automatically generated**

**CONCLUSION: Thus, we implemented programs on Constructor and Destructor.**