**Name :Omkar A Ghogale**

**Roll no:07**

**Div:D**

**Branch:IT**

**Experiment No:03**

**Experiment 3**

**Aim:** Write a program to print the area of a rectangle by creating a class named 'Area' having two methods.First method named as 'setDim' takes length and breadth of rectangle as parameters and the second method named as 'getArea' returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard.

**Theory :**

**Methods in java :**

A method is a collection of statements that perform some specific task and return the result to the caller. A method can perform some specific task without returning anything.

Methods are **time savers** and help us to **reuse** the code without retyping the code.

In general, method declarations has six components :

● **Modifier**-: Defines **access type** of the method i.e. from where it can be accessed in

your application. In Java, there 4 types of access specifiers.

○ public: accessible in all classes in your application.

○ protected: accessible within the class in which it is defined and in its **subclass(es)**

○ private: accessible only within the class in which it is defined.

○ default (declared/defined without using any modifier): accessible within the same class and package within which its class is defined.

* + **The return type**: The data type of the value returned by the method or void if does

not return a value.

* + **Method Name**: the rules for field names apply to method names as well, but the

convention is a little different.

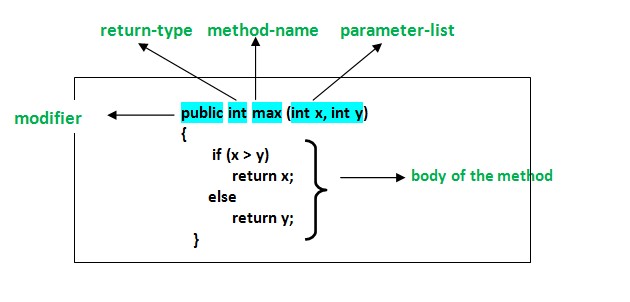
* + **Parameter list**: Comma separated list of the input parameters are defined, preceded with their data type, within the enclosed parenthesis. If there are no parameters, you

must use empty parentheses ().

* + **Exception list**: The exceptions you expect by the method can throw, you can specify

these exception(s).

* + **Method body**: it is enclosed between braces. The code you need to be executed to perform your intended operations.



**Conclusion:** Thus studied functions in java.

**Questions:**

Q.1 Explain method overloading in java.

Method overloading **is a concept** that allows to declare **multiple methods with same name but different parameters in the same class.**

Java supports method overloading and always occur in the same class(unlike method overriding).

Method overloading is one of the ways through which java supports polymorphism. Polymorphism is a concept of object oriented programming that deal with multiple forms. We will cover polymorphism in separate topics later on.

Method overloading can be done by **changing number of arguments** or by **changing the data type of arguments** If two or more method have same name and same parameter list **but differs in return type** can not be overloaded.

There are two different ways of method overloading.

1. Different datatype of arguments
2. Different number of arguments

Q.2 Explain any five string functions.

**String length()**: The Java String length() method tells the length of the string. It returns count of total number of characters present in the String.

**String compareTo()**: The Java String compareTo() method compares the given string with current string. It is a method of*‘Comparable’* interface which is implemented by String class. Don’t worry, we will be learning about String interfaces later. It either returns positive number, negative number or 0.

**String concat() :**The Java String concat() method combines a specific string at the end of another string and ultimately returns a combined string. It is like appending another string.

**String IsEmpty()** : This method checks whether the String contains anything or not. If the java String is Empty, it returns true else false

**String toLowerCase()** : The java string toLowerCase() method converts all the characters of the String to lower case.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

.

