

# **Assignment 1**

**Saad Ahmed Salim**  
**1712299042**  
**Section-2**

**6 December, 2020**

**CSE-215**

**Dr. Ziaul Hossain**

## Section 1

(2) A class can have three kinds of members.

- fields
- methods
- nested classes and nested interfaces

Fields are data variables which determine the status of the class or an object. Example: If we have a Box class, our fields will be width, height, depth etc.

Methods allow us to modify the status of an object or access the value of the data members. Example: We can access the fields of Box class and change the value.

We can also do nested classes and interfaces in Java.

## Section - 2

(5) Below code - 1 contains correct overloading code.

Main condition of overloading:

- 1st we have to check the number of parameters. If they are same, we have to check the type of parameters. If both are same, it is not possible to run.

So, code - 1 have same number of parameters. But the types are different. So, it allows method overloading.

But, code - 2 won't work because the numbers of parameters and the types of parameters are same. It will show an error.

Section-3

(7) I have created two classes inside default package of LabRoom project. LabRoomDemo.java is our main class and LabRoom.java is our driver class. I will show my code of LabRoom.java in below:

```

public class LabRoom {
    private int chair;
    private int table;
    private int computer;

    public LabRoom() {
        chair = 0;
        table = 0;
        computer = 0;
        // Default constructor set; if there is no
        // value it will show 0 output for the fields
    }

    public LabRoom(int chair, int table, int computer) {
        setChair(chair);
        setTable(table);
        setComputer(computer);
    }
}

```

```
public void setChairc(int chairc){  
    this.chairc = chairc;  
}  
public int getChairc(){  
    return chairc;  
}  
public void setTable(int table){  
    this.table = table;  
}  
public int getTable(){  
    return table;  
}  
public void setComputerc(int computerc){  
    this.computerc = computerc;  
}  
public int getComputerc(){  
    return computerc;  
}  
public void print(){  
    System.out.println("Lab Room has : "  
        + " Chairc : " + getChairc()  
        + ", Table: " + getTable()  
        + " and Computerc : " + getComputerc());  
}
```

```
public void addChair (int number) {  
    this.chair += number;  
}  
  
public void addTable (int number) {  
    this.table += number;  
}  
  
public void addComputer (int number) {  
    this.computer += number;  
}  
  
public void removeChair (int number) {  
    this.chair -= number;  
}  
  
public void removeTable (int number) {  
    this.table -= number;  
}  
  
public void removeComputer (int number) {  
    this.computer -= number;  
}
```

Now, the code of LabRoomDemo.java with main class is shown below:

```
public class LabRoomDemo {  
    public static void main (String [] args) {  
        LabRoom labroom1 = new LabRoom();  
        // it will print default constructor  
        // if we print  
        // now I am creating object of  
        // LabRoom  
        LabRoom labroom2 = new LabRoom(20,20,20);  
        LabRoom labroom3 = new LabRoom(20,18,15);  
        labroom2.print();  
        labroom3.print();  
        // now subtraction  
        int addChair = labroom2.getChair()  
                    - labroom3.getChair();  
        int addTable = labroom2.getTable()  
                    - labroom3.getTable();  
        int addComputer = labroom2.getComputer()  
                    - labroom3.getComputer();  
    }  
}
```

```

System.out.println ("Need to Subtract"
+ "LabRoom2 - LabRoom3");

System.out.println ("Chairs: " + addChair
+ " Tables: " + addTable + " Computers: "
+ addComputer);

// now adding all elements in LabRoom3
System.out.println ("After adding all"
+ "elements in LabRoom3: ");

labroom3.addChair (addChair);
labroom3.addTable (addTable);
labroom3.addComputer (addComputer);

// labroom2.print();
labroom3.print();

}

```

{}

Output of the Code will be :

LabRoom has Chairc: 20, Table: 20 and - Computerc: 20  
 LabRoom has Chairc: 20, Table: 18 and Computerc: 15  
 Need to Subtract LabRoom2 - LabRoom3:  
 Chairs: 0 Tables: 2 Computers: 5  
 After adding all elements in LabRoom3:  
 LabRoom has Chairc: 20, Table: 20 and Computerc: 20