OBJECT ORIENTED PROGRAMMING
LAB# 06 TASKS
Understanding the Concept Of Overloading

Ex #1: Write a java program consisting of two static methods.

First method will add 2 integer numbers.

Second method will add 3 integer numbers.

Class:

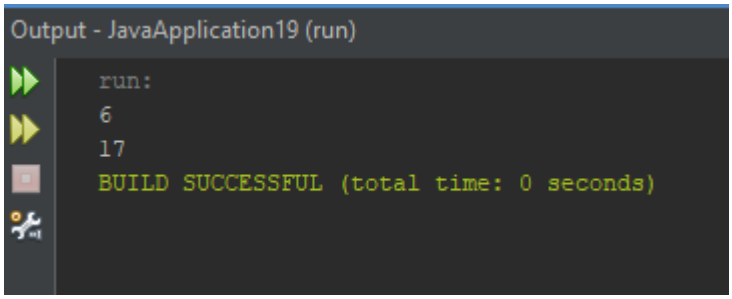
```
package javaapplication19;
/**
 *
 * @author user
 */
public class Calculate {
   int sum;

   public void add (int a,int b){
      sum = a+b;
      System.out.println(sum);
   }
   public void add (int a,int b,int c)
   {
      sum = a+b+c;
      System.out.println(sum);
   }
}
```

Main:

```
package javaapplication19;

/**
 *
```

```
 * @author user
 */
import java.util.Scanner;
public class JavaApplication19 {

   /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {
      // TODO code application logic here
      Calculate S1 =new Calculate();
       S1.add(2,4);
      S1.add(4,5,8);
   }

}
```

Output:



Ex # 2: Write a Java program to display StudentID, Name and Grade using method overloading.

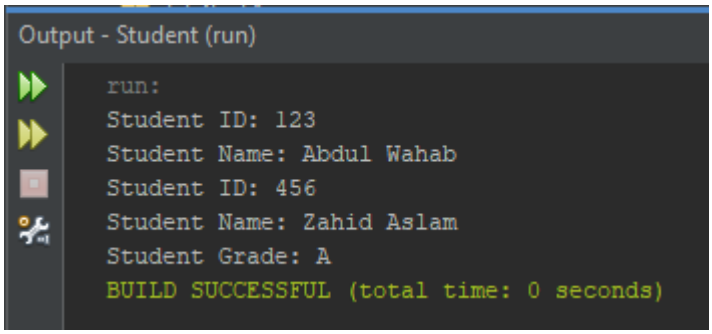
(Hint: Use two parameters in first method and 3 parameters in second method.)

Solution:

```
package student;

public class Student {

   public static void displayDetails(int id, String name) {
```

```
        System.out.println("Student ID: " + id);
        System.out.println("Student Name: " + name);
    }

    // Method to display student details with 3 parameters
    public static void displayDetails(int id, String name, char grade) {
        displayDetails(id, name); // Reuse the first method to display ID and name
        System.out.println("Student Grade: " + grade);
    }

    public static void main(String[] args) {
        displayDetails(123, "Abdul Wahab");
        displayDetails(456, "Zahid Aslam", 'A');
    }
}
```

Output:



Task:5

Class:

```
package task5.lab6;
/**
 *
 * @author user
 */
public class Rectangle {

    private double length;
    private double breadth;
```

```
    public Rectangle(double length, double breadth) {
        this.length = length;
        this.breadth = breadth;
    }

    public double getLength() {
        return length;
    }

    public double getBreadth() {
        return breadth;
    }

    public Rectangle operatorPlus(Rectangle other) {
        double newLength = this.length + other.getLength();
        double newBreadth = this.breadth + other.getBreadth();
        return new Rectangle(newLength, newBreadth);
    }
}
```

Main:

```
package task5.lab6;
/**
 *
 * @author user
 */
public class Task5Lab6 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
```

```
        Rectangle r1 = new Rectangle(4, 5);

        Rectangle r2 = new Rectangle(3, 6);

        Rectangle r3 = r1.operatorPlus(r2);

        System.out.println("Length of rectangle 1: " + r1.getLength() + ", Breadth of rectangle 1: " +
r1.getBreadth());

        System.out.println("Length of rectangle 2: " + r2.getLength() + ", Breadth of rectangle 2: " +
r2.getBreadth());

        System.out.println("Length of rectangle 3: " + r3.getLength() + ", Breadth of rectangle 3: " +
r3.getBreadth());

    }

   }
```

Output: