Program Name	Description
BSLab9a.java	Write a Java program which prints "Hello World" to the screen 5 times. The main method should contain a loop which iterates for 5 times and invoke a method call sayHello() each time. The method sayHello() should only display the message to the screen.

Output

```
Hello World
Hello World
Hello World
Hello World
Hello World
```

Program Name	Description
BSLab9b.java	Write a Java program which invokes two methods: printOdds() – Method which prints odd numbers between 1 and 20 printEvens() – Method which prints even numbers between 1 and 20 **NOTE:** The methods should use a loop to print the numbers

Output

```
Odd Numbers.: 1 3 5 7 9 11 13 15 17 19
Even Numbers: 2 4 6 8 10 12 14 16 18 20
```

Program Name	Description
BSLab9c.java	Write a Java program that takes two numbers from a user, sends the numbers to a method called **multNumbers**, which **returns** the result of the first number multiplied by the second one. The result should be displayed back to the user.

Output

```
1st Number: 31
2nd Number: 12

31 * 12 = 372
```

NOTE: The numbers above are just to illustrate the output of the program !!!

Program Name	Description
BSLab9d.java	Write a Java program that takes a name and an age from user, sends the data to a method called **checkPerson**, which **receives** the parameters (name and age), classifies the user according to the user's age and displays the classification back to the user: - Children (age less or equal than 12) - Adult (age between 13 and 50) - Senior (age greater 50)

Output

```
Enter your name: Aoife

Enter your age.: 27

Aoife ( 27 years old ) is classified as: Adult
```

NOTE: The values are to illustrate how the program should work

Program Name	Description
BSLab9e.java	Write a Java program which takes in an input from user and validates the input using a method called validateInput(). The method should receive the input as a parameter and return true if the input is a number or false if it is not. If the input is not a number, the program should ask the user for a new input.

Output

```
Please, enter a number: a

INVALID INPUT, Try again...

Please, enter a number: 10

Valid input, thank you.
```

NOTE: The values are to illustrate how the program should work

Program Name	Description
BSLab9f.java	Write a Java program that takes a speed in kilometres / hour (Km/h) from user and converts to Miles / hour (MPH). You program should have a method called **convSpeed** that takes the speed entered by the user, converts it and **returns** the result. The result should be printed back to the user. **1 Kilometre = 0.621371 Mile**

Output

```
Speed in Km/h: 100

Speed in MPH: 62.13
```

NOTE: The values are to illustrate how the program should work

Program Name	Description
BSLab9g.java	Write a Java program which: 1. Generates a list with 10 random numbers between 1 and 50 2. Displays the list The program should have **two** methods: ***populateArray()*** – Method that generates a list of 10 random numbers and return the list to the main method. ***displayArray()*** – Method that received the populated array as a parameter and displays to the screen all the elements of it.

Output

```
List: 27 9 25 41 10 2 5 35 20 22
```

Program Name	Description
BSLab9h.java	Write a Java program that operates like a calculator. Your program should ask a user for two numbers, and then show a menu of options after the numbers have been entered, each operation should be done in a **different** method. Each method should take the two numbers as parameters, and **return** the result of the operation, the result should also be printed back to the user.

Output

```
Enter the 1st number: 9

Enter the 2nd number: 27

---- Menu ----
 a - Add
 b - Subtract
 c - Divide
 d - Multiply
--------------

Enter one option: a

9 + 27 = 36
```

NOTE: The value 9, 27, a and 36 are to illustrate who the program should work

Sample Exam Question

A. Write a Java program that takes a input from user, validates it through a method called **validNumber()** and if the input is a valid number, displays to back to the user whether the number is an odd number or an even number, the program should keep taking numbers from user until the user presses **X**.

- Assume that the method **validNumber()** exists and return a **boolean**
- Assume that the class and main method have been correctly declared

The output of the program should be:

```
Please, enter a number ( or X to finish ): __
```

If the number entered is **NOT** valid, then displays the following message:

```
Invalid number !  Try again
```

If the number enter is valid, then display **ONE** of the following messages:

For **odd** numbers: **The number entered is an ODD number**
For **even** numbers: **The number entered is an EVEN number**

B. **Using full Java code,** write a program that contain the following methods:

- **meanNumb**, which receives two integers (**num1** and **num2**) and returns the mean between of the numbers
- **sumNumb**, which receives two integers (**num1** and **num2**) and returns the sum of the numbers

The program should consider two random numbers between **1 and 100 (inclusive)** to be used in all methods above, and the output of the program should be:

```
Number 1: X
Number 2: Y

Mean: A
Sum: B
```

X and **Y** are the random numbers, **A** is the result of the meanNumb and **B** is the result of the sumNumb.

C. Write a Java program that displays the following menu to a user:

```
( A ) - Add new Student
( S ) – Show Student
( M ) – Modify Student
( X ) – Quit

Enter an option: ___
```

The program should also accept input from the user, and execute a different method for each choice (**using switch statement**), according to the following:

If the user types **A**, the program should run a method called **addStudent()**
If the user types **S**, the program should run a method called **showStudent()**
If the user types **M**, the program should run a method called **modifyStudent()**
If the user types **X**, the program should finish the program

NOTE: All methods (addStudent, showStudent and modifyStudent) along with the class and main method are already declared. No checking or validation is required