**National University of Computer & Emerging Sciences, Karachi**



**Computer Science Department Spring 2023, Lab Manual – 02**

|  |  |
| --- | --- |
| **Course Code: CL-1004** | **Course : Object Oriented Programming Lab** |
| **Instructor(s) :** | **Abeer Gauher, Hajra Ahmed, Shafique Rehman** |

LAB - 2

Operators, Control Structures, loops & Math Library Functions in Java

# CONTENTS:

* Operators
* Control structures.
  + If statements
  + If else statements
  + If else if statements
  + For loop
  + While loop
  + Do while loop
* Java Math class

# Operators

Operators are classified into different types shown below:

**postfix (x++, x--)**

**unary**

**prefix (++x,- -x , ! x)**

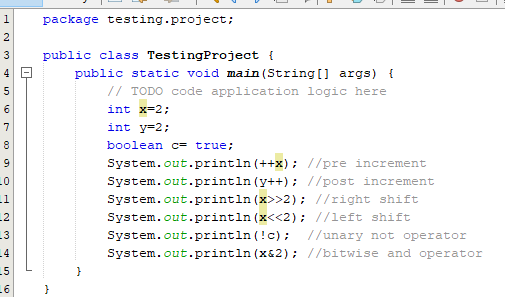
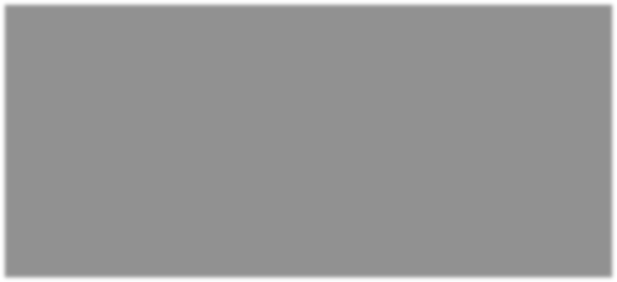
**artithmetic ( +, -, \*, /, % ) shift (>>, <<)**

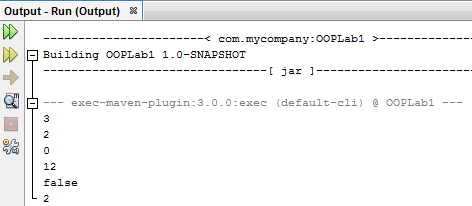
**operators**

**relational (>, <, >=, <= ,== , != )**

**bitwise ( &, | ) logical ( && , || ) ternary (? :)**

**Assignment (=)**





**Ternary operator:**

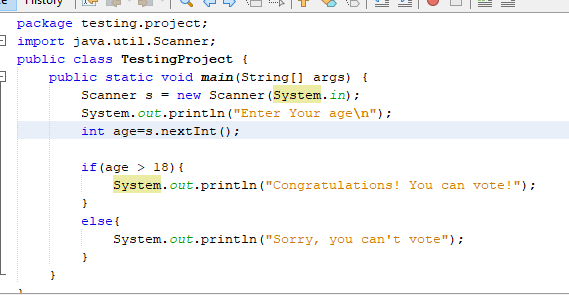
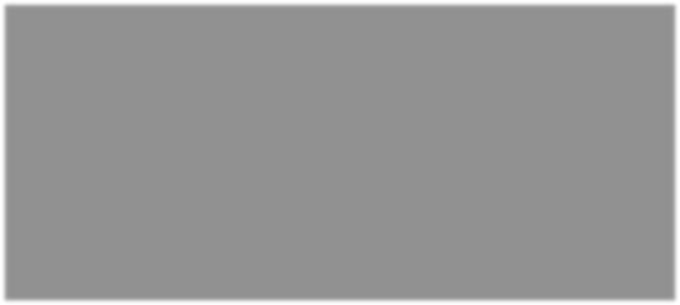
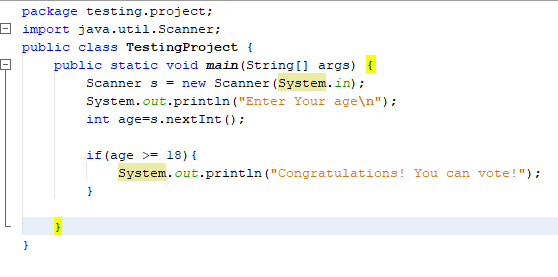
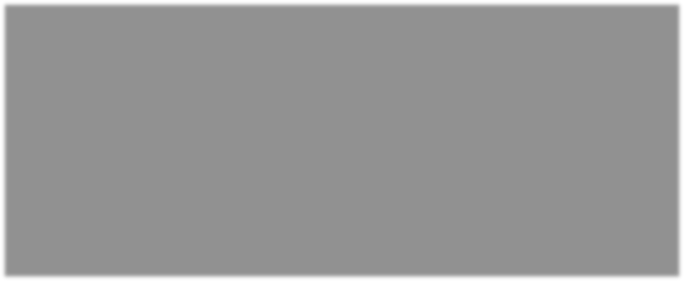
bool data= age>18? “can vote”: “cannot vote”

# Control structures:

Java compiler executes the code from top to bottom. The statements in the code are executed according to the order in which they appear. However, [Java](https://www.javatpoint.com/java-tutorial) provides statements that can be used to control the flow of Java code. Such statements are called control flow statements. It is one of the fundamental features of Java, which provides a smooth flow of program.

* Decision making (if statements, if else statements, switch statements etc.)
* Loop statements (while, do while, for, foreach)
* Jump statements (break, continue)

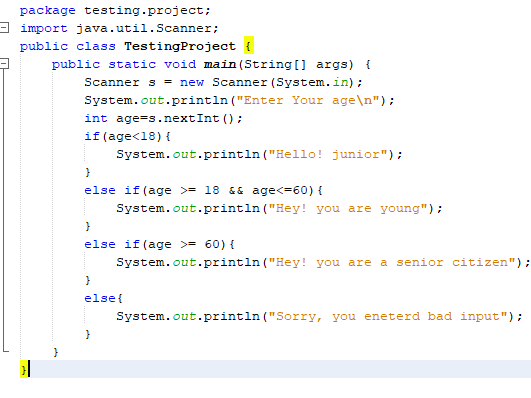
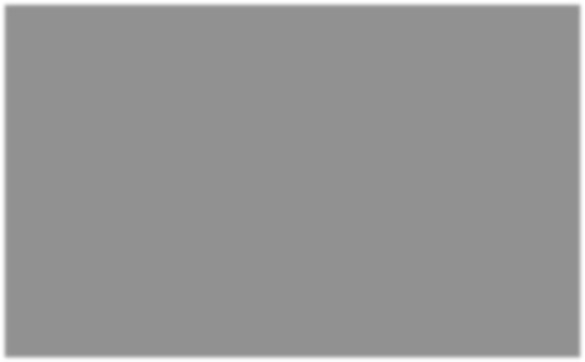
# Decision making statements:



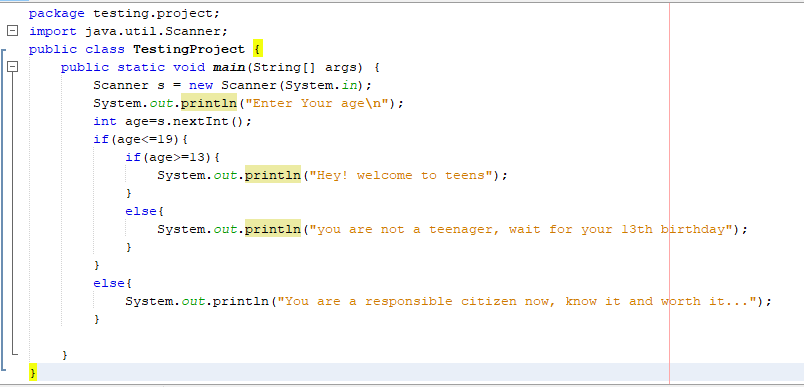
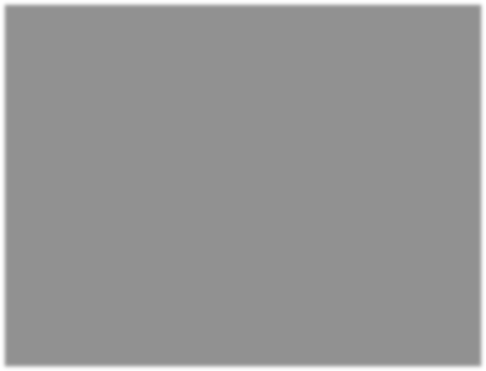
In Java, the "if" statement is used to evaluate a condition. The control of the program is diverted depending upon the specific condition. The condition of the If statement gives a Boolean value, either true or false. In Java, there are four types of if-statements, i.e. if statement, if-else statements, if else-if ladder, nested if statements.

*Figure 1: if statement*

*Figure 2:if else statements*

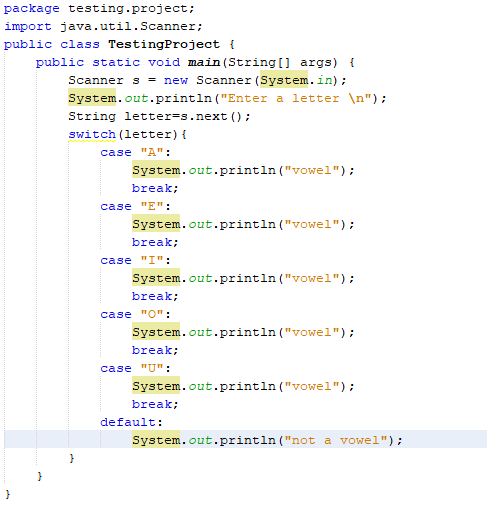
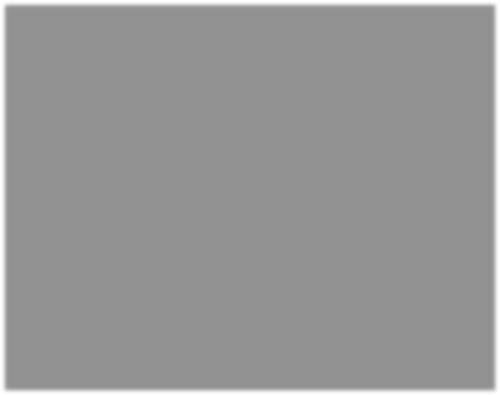


*Figure 3:if else if statements*



*Figure 4: nested if statements*

If an if statements is placed with in the else statement then it is termed as if-else if ladder. Consider it a DIY exercise.



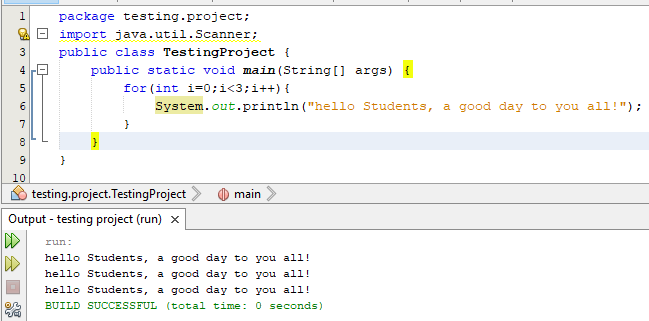
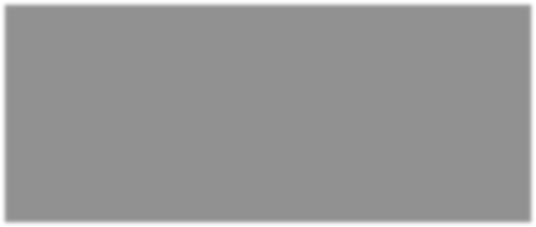
*Figure 5: Switch case*

We will cover break statement in next sections.

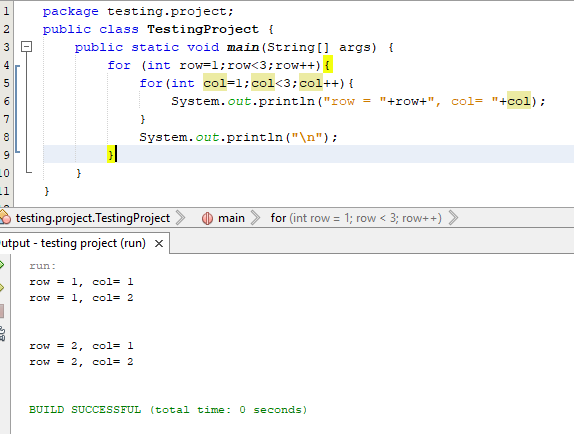
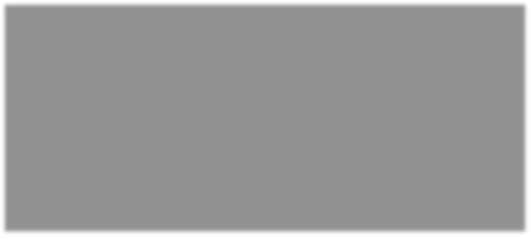
# Loops:

In programming, sometimes we need to execute the block of code repeatedly while some condition evaluates to true. However, loop statements are used to execute the set of instructions in a repeated order. The execution of the set of instructions depends upon a particular condition.

In Java, we have three types of loops namely for loop, while loop and do while loop.

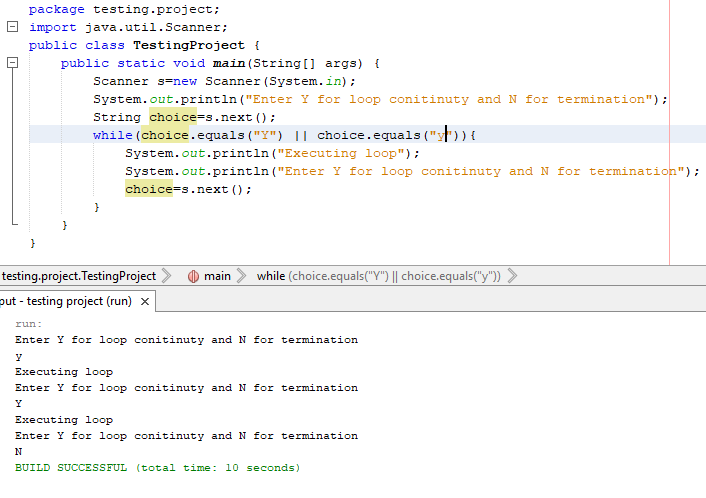


*Figure 6: for loop*



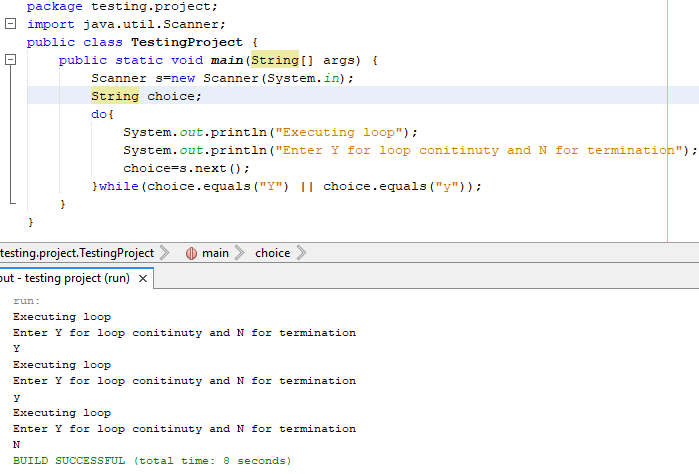
*Figure 7:nested for loop*

While loop will check the condition first and then execute the statements if condition is true.



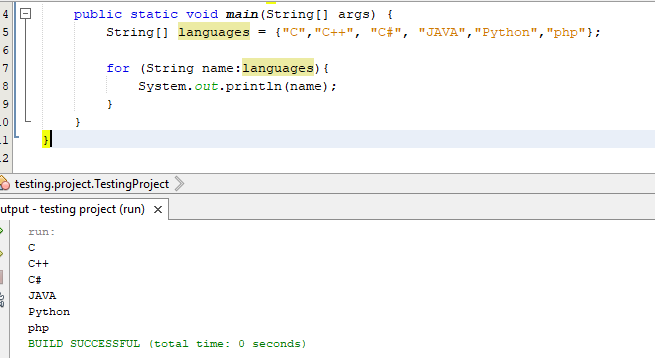
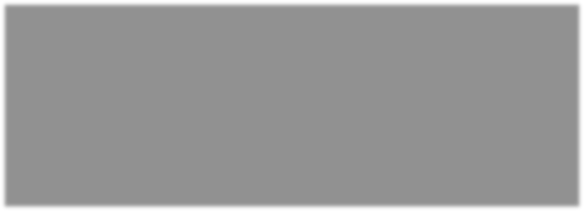
*Figure 8: while loop*

Do while will execute loop body once and then check the condition.



*Figure 9: do while loop*

For-each is another array traversing technique like for loop, while loop, do-while loop introduced in Java5. It has a slightly different syntax than for loop shown below:

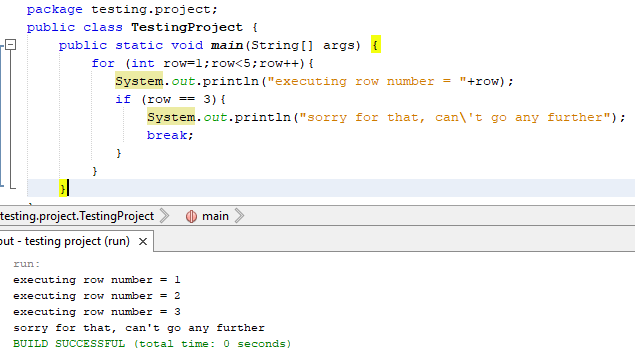
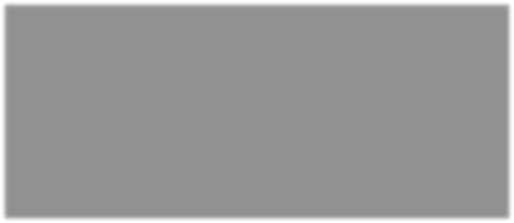


*Figure 10:foreach loop*

# jump statements:

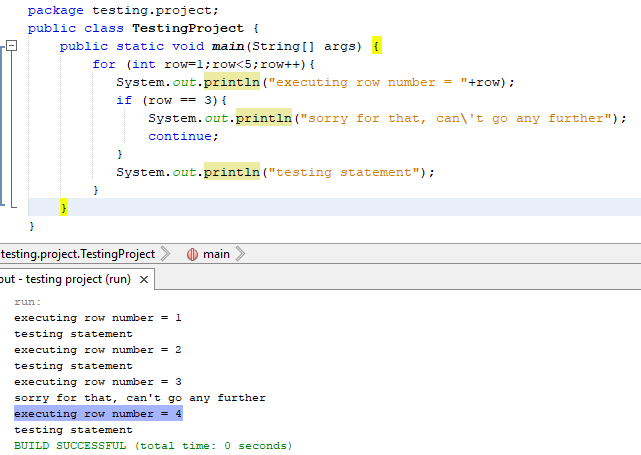
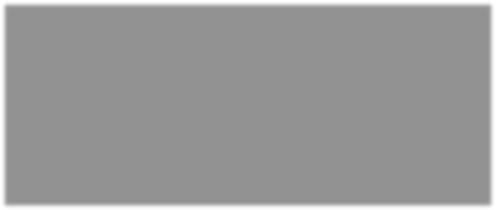
The break and continue statements are jump statements that are used to bypass some loop statements or finish the loop without verifying the test expression. These statements can be inserted into any loop, including for, while, and do-while loops.

The **break** statement in java is used to terminate from the loop immediately. When a break statement is encountered inside a loop, the loop iteration stops there, and control returns from the loop immediately to the first statement after the loop. Basically, break statements are used in situations when we are not sure about the actual number of iterations for the loop, or we want to terminate the loop based on some condition.



*Figure 11: break statement*

The **continue** statement in Java is used to skip the current iteration of a loop. Generally, they are used in the situations when we want to continue the loop but do not want the remaining statement after the continue statement to executed.



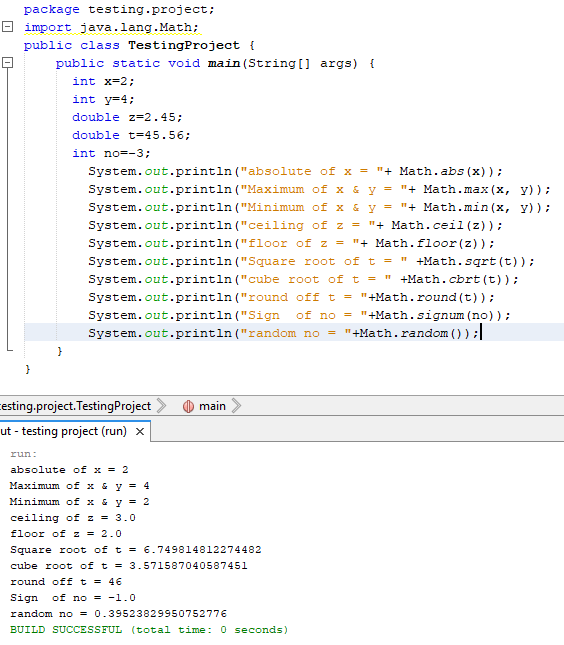
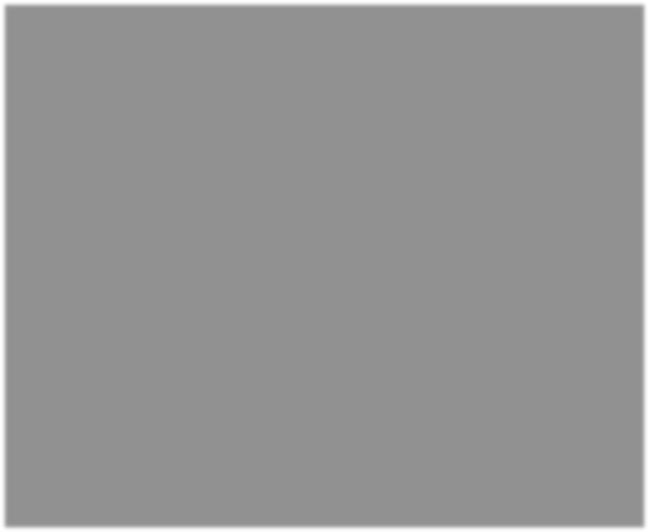
*Figure 12: continue statement*

There is no go to statement in Java but still goto is a reserved keyword. If they want to add it to the later versions. For now, the break statements use labels to jump to a specific line of code.

# Java Math Class:

Java Math class provides several methods to work on math calculations like min(), max(), avg(), sin(), cos(), tan(), round(), ceil(), floor(), abs() etc.

|  |  |
| --- | --- |
| **Function** | **Description** |
| **Math.abs(x)** | It will return the Absolute value of the given value. |
| **Math.min(x,y)** | It returns the Largest of two values |
| **Math.max(x,y)** | It is used to return the Smallest of two values. |
| **Math.round(x)** | It is used to round of the decimal numbers to the nearest value. |
| **Math.sqrt(x)** | It is used to return the square root of a number. |
| **Math.cbrt(x)** | It is used to return the cube root of a number. |
| **Math.pow(x,y)** | It returns the value of first argument raised to the power to second argument. |
| **Math.signum(x)** | It is used to find the sign of a given value. |
| **Math.ceil(x)** | Compute ceiling of a number (round up forward to nearest decimal place) |
| **Math.floor(x)** | Compute floor of a number (round off backward to nearest decimal place) |
| **Math.random(x)** | generate a positive random number between 0 to 1. |
| **Math.sin(x)** | It is used to return the trigonometric Sine value of a Given double value. |
| **Math.log(x)** | It returns the natural logarithm of a double value. |



*Figure 13: math library functions*

# LAB TASKS:

**TASK – 01:**

Your task is to design and implement an Invoice application in Java that will print the formatted invoice. Your program starts by inputting sale line items and their prices. It should then calculate the sub-total, tax, discount, and the final bill for a customer. Assume that you are designing this application to be used by a cashier at the POST (Point Of Sales Terminal). So, despite of terminating the program after every sale, use loop to ask user if she wants to continue another sale (for the sake of simplicity assume that each sale consists of 5 items only). Program will terminate if the user enters ’y’ or ’Y’.

**Here is a sample interaction:**

Enter items (first item’s name and then price):

Chips 50

Hand Wash 200

Canned Almonds 125

Pickle 145

Apple juice 70



Sample Output

|  |  |  |
| --- | --- | --- |
|  | INVOICE |  |
| Subtotal: |  | $590.00 |
| Discount percent: |  | 10 |
| Discount amount: |  | $59.00 |
| Total before tax: |  | $531.00 |
| Sales tax: |  | 5 |
| Invoice total: |  | $557.55 |
| Continue(y/n): |  |  |

# TASK – 02:

# Write a program that reads a set of integers from 1 to 100, and then finds and prints the sum of the even and odd integers separately.

# TASK – 03:

Write a Java program that asks a number from user and display if its divisible by 3 or 5 and by both.

# TASK – 04:

Write a program which calculates the grace marks for a student in Object oriented programming course using switch case. The user should enter the grade obtained by the student and the number of assignments he has failed in.

* + - If the student gets A grade and the number of assignments, he failed in is greater than 3, then he does not get any grace. If the number of assignments he failed in is less than or equal to 3 then the grace is of 5 marks.
    - If the student gets B grade and the number of assignments, he failed in is greater than 2, then he does not get any grace. If the number of assignments he failed in is less than or equal to 2 then the grace is of 4 marks.
    - If the student gets C grade and the number of assignments, he failed in is greater than 1, then he does not get any grace. If the number of assignments he failed in is equal to 1, then the grace is of 5.

Display the appropriate output in each case.

# TASK – 05:

Create a menu driven transactions’ processing unit in a bank as long as user enters correct input. Initially, take the username and password as input from user. If that username and password is correct then the user will log in to the system and can perform any transaction as per the menu shown below.

Main Menu \*\* Welcome to Bank of Pakistan\*\*

1. Deposit Money
2. Withdraw Amount
3. Account status Select your choice:

(After completing the selected transaction) Do you want to continue? [y/Y] (goes to Main Menu, if y/Y is pressed)

Depending upon the user’s choice, perform the transaction and display the remaining account balance along with the owner’s username.