NUST

*Defining futures*

NCDC

NUST CHIP DESIGN CENTRE

# Digital Design Verification

## Assignment # 01

## Conditionals

**Release: 1.1**

**Date: 27-April-2024**

**NUST Chip Design Centre (NCDC), Islamabad, Pakistan**

## Revision History

| Revision Number | Revision Date | Nature of Revision | Approved By |
|---|---|---|---|
| 1.0 | 2/05/2024 | Complete manual | Dr. Abid |
| 1.1 | 30/07/2024 | Revision in manual | Hira Sohail |

Azad’s Law: The one facilitating cheating will be punished with 0 marks]

The objective of this assignment is to enable students to answer following questions:

- How to use flow charts to devise algorithms for solving problems?
- What sort of problems require decision making?
- How to program computers to solve such problems?
- How does C/C++ support decision making?
- What kind of Machine Learning/AI problems need decision making and how can we leverage the knowledge built so far to solve such problems

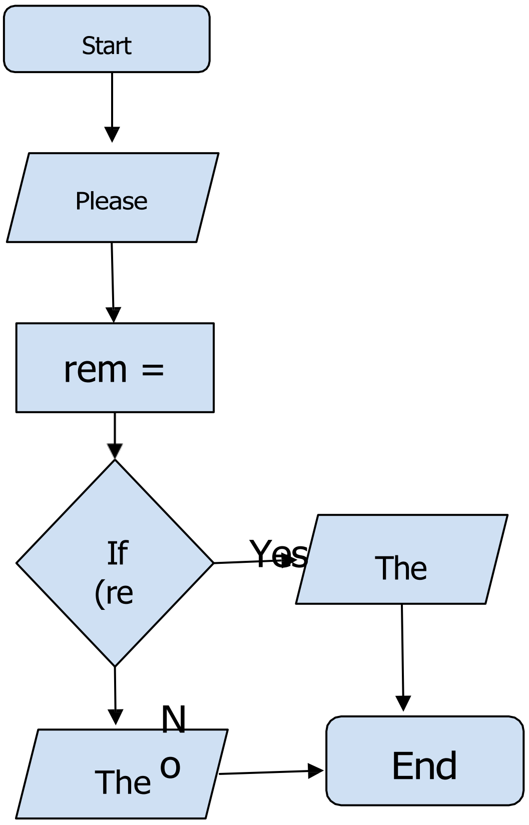
## 1. Flow Charts for Structured Problem Solving

As introduced in theory class, flow charts are one of the simplest tools used for structured problem solving. Let us explain the usage and symbols using a simple case study.

**Problem:** Take a number from the user and

find out whether it is even or odd?

**Solution:**



```
#include<iostream>

using namespace std;


int main()

{

            int number = 0;


            cout<<"Please  enter  the
number"<<endl;

            cin>>number;


            int rem = number % 2;


            If (rem == 0)

            {

                        cout << "The number is
 even" <<endl;

            }

            else

            {

                                    cout << "The
            number is odd" <<endl;

            }
```

## 2. Decision Making in C/C++

There are two constructs that are used for decision making in C/C++. These include:

- If-else if -else
- Switch

The above example can be solved using a switch statement as shown below.

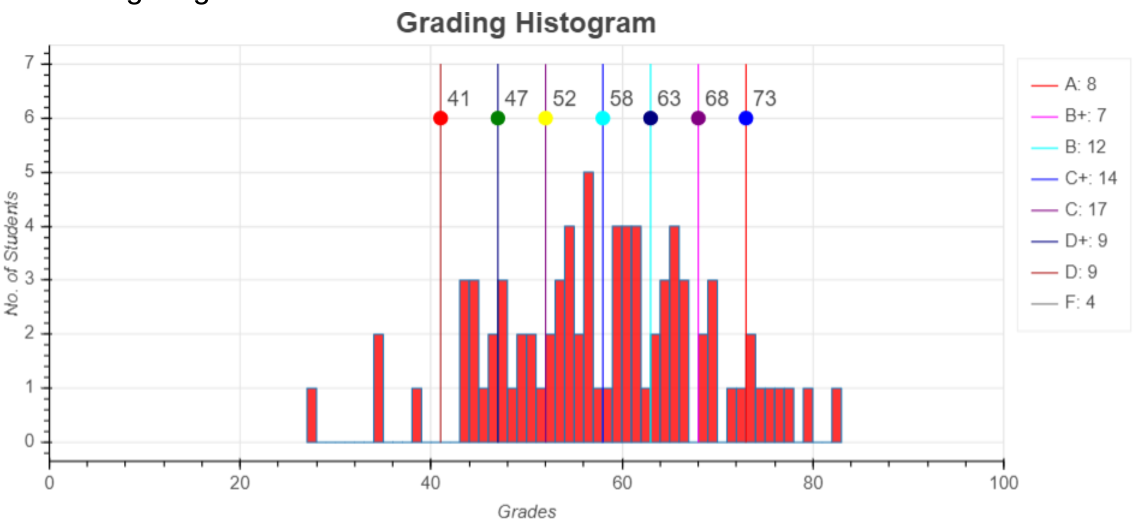
```
#include<iostream>

using namespace std;


int main()

{

        int number = 0;


        cout<<”Please enter the number”<<endl;

        cin>>number;


        int rem = number % 2;


        If (rem == 0)

        {

                cout << “The number is even” <<endl;

        }

        else

        {

                        cout << “The number is odd” <<endl;

        }


        return 0;

}
```

```
#include<iostream>

using namespace std;


int main()

{

        int number = 0;


        cout<<”Please enter the number”<<endl;

        cin>>number;


        int rem = number % 2;

        switch (rem)

        {

                case 0:

                        cout << “The number is even” <<endl;

                        break;

                case 1:

                        cout << “The number is odd” <<endl;

                        break;

                default:

                        cout << “Invalid case” <<endl;

                        break;

        }


        return 0;

}
```

**Figure.** if-else vs. Switch

## TASK - 01:

Qalam is a system used at NUST to calculate the grade of students using a relative grading method. Typically, a bell curve is formed, and a number of thresholds are used to decide the grades. The decision boundaries are shown. As an example, if the marks are greater than 73, then it is an A. Likewise if the marks are greater than 68, then it is B+ and so on and so forth. Write a program that takes the grades as input and assigns grades based on the decision boundaries mentioned in the diagram below.



***Flow Chart:*** 10 minutes ***Coding*** **:** 15 minutes

A few samples run on the program

1. Please enter the grade: 74

   Congrats! You got an A grade

2. Please enter the grade: 52

   You got a C grade

## Decision Making Applications in Machine Learning/AI

Typically, there are two major classes of problems where we apply Machine Learning: prediction and classification. Both of these problems involve decision making based on some features of the data.

## TASK - 02 Prediction:

Imagine you have to write a program which predicts your electricity bill based on the historic data of different consumers' billing patterns as shown below.

| S.No. | Number of Units | Electricity Bill (PKR) |
|---|---|---|
| 1. | 200 | 3600 |
| 2. | 100 | 1800 |
| 3. | 250 | 7500 |
| 4. | 350 | 10500 |
| 5. | 500 | 20,000 |
| 6. | 400 | 16,000 |

## TASK - 03 Classification:

Consider the following data which classifies whether a person is fit or not based on his age, weekly pizzas intake and daily morning exercise routine. Write a code which takes input from the user the age, number of pizzas eaten per week and whether he is doing daily morning exercise or not and then classify whether he is 'Fit' or 'Unfit'.

| S.No. | Age (Years) | Number of Pizzas Per Week | Daily Morning Exercise | Fit/Not Fit |
|---|---|---|---|---|
| 1. | 25 | 7 | Yes | UnFit |
| 2. | 35 | 1 | No | UnFit |
| 3. | 40 | 3 | Yes | Fit |
| 4. | 20 | 2 | No | Fit |

| | | | | |
|---|---|---|---|---|
| 5. | 50 | 0 | No | UnFit |

***Flow Chart*** **:** 20 minutes ***Coding*** **:** 20 minutes

A few samples run on the program

```
1.    Please enter the age of the person (years) : 28
      Please enter how many pizzas he eat per week : 10
      Please mention whether he exercises daily in the morning: Y
      The person is ‘Unfit’

2.    Please enter the age of the person (years) : 23
      Please enter how many pizzas he eat per week : 1
      Please mention whether he exercises daily in the morning: N
      The person is ‘Fit’

3.    Please enter the age of the person (years) : 43
      Please enter how many pizzas he eat per week : 1
      Please mention whether he exercises daily in the morning: N

      The person is ‘UnFit’

4.    Please enter the age of the person (years) : 53
      Please enter how many pizzas he eat per week : 3
      Please mention whether he exercises daily in the morning: Y

      The person is ‘Fit’
```

## TASK - 04

Newton’s Gravitational Law is used to compute the gravitational force between two masses using the following equation.

$$F = k\ (M_1 * M_2)/d^2$$

where

$M_1$ = mass of body 1

$M_2$ = mass of body 2

d = distance between the two bodies

$k$ = constant whose value is specified as $6.67*10^{-8}$ dyn. $cm^2$ / $g^2$

Take the masses of objects as inputs from the user as well as the distance between them and calculate the gravitational force between them.

## Submission:

Please add all the flow charts neatly in your report.

Attach .c files of all the tasks along with the screenshots of outputs on LMS in a proper report.