Lab Task 01

Submitted By:

Wasif Mehmood

Submitted To:

DR Zakira Inayat

Reg No:

20pwbcs0769

Department:

CS&IT (Section B)

Date 25th April,2021.

Q1. What is an algorithm? Why algorithms are important?

# ALGORITHM:

An algorithm is a set of step-by-step procedures, or a set of rules to follow, for completing a specific task or solving a particular problem. Algorithms are all around us. The recipe for baking a cake, the method we use to solve a long division problem, and the process of doing laundry are all examples of an algorithm. Here’s what baking a cake might look like, written out as a list of instructions, just like an algorithm:

* Preheat the oven
* Gather the ingredients
* Measure out the ingredients
* Mix together the ingredients to make the batter
* Grease a pan

# Example:

## Algorithm of Grades:

Steps are given below:

Start

Input Marks1, Marks2, Marks3, Marks4

Grade= (Marks1+Marks2+Marks3+Marks4)/4

If (Grade<50) then

Print “Fail”

Else

Print “Pass”

End if

Stop

# Algorithms Used in Computer Science:

Algorithms are used in every part of computer science. They form the field's backbone. In computer science, an algorithm gives the computer a specific set of instructions, which allows the computer to do everything, be it running a calculator or running a rocket. Computer programs are, at their core, algorithms written in programming languages that the computer can understand. Computer algorithms play a big role in how social media works: which posts show up, which ads are seen, and so on. These decisions are all made by algorithms. Google’s programmers use algorithms to optimize searches, predict what users are going to type, and more. In problem-solving, a big part of computer programming is knowing how to formulate an algorithm.

# THE IMPORTANCE OF ALGORITHMS:

You are planning to build a big house but at the same time, you are not sure whether the resources that you have are enough. What will you do? You will define a work plan that will ensure you spend the little resources available at your disposal to finish the building. Secondly, you are planning to travel several miles away but there is very little time available. It is quite obvious that you will get the shortest or fastest route that will get you to your destination.

When it comes to [computer programming](https://www.technotification.com/category/coding), algorithms work in a similar manner. In layman’s language, an algorithm can be defined as a step by step procedure for accomplishing a task. In the world of programming, an algorithm is a well-structured computational procedure that takes some values as input some values as output.

Algorithms give us the most ideal option of accomplishing a task. Here is some importance of algorithms in computer programming.

## 1. To improve the efficiency of a computer program

In programming, there are different ways of solving a problem. However, the efficiency of the methods available vary. Some methods are well suited to give more accurate answers than others. Algorithms are used to find the best possible way of solving a problem. In doing so they improve the efficiency of a program.

When it comes to programming, efficiency can be used to mean different things. One of them is the accuracy of the software. With the best algorithm, a computer programme will be able to produce very accurate results.

Another way of looking at the efficiency of software is speed. An algorithm can be used to improve the speed at which a program executes a problem. A single algorithm has the potential of reducing the time that a program takes to solve a problem.

## 2.Proper utilization of resources

A typical computer has different resources. One of them is [computer memory](https://en.wikipedia.org/wiki/Computer_memory). During the execution phase, a computer program will require some amount of memory. Some programs use more memory space than others. The usage of computer memory depends on the algorithm that has been used.

The right choice of an algorithm will ensure that a program consumes the least amount of memory. Apart from memory, and the algorithm can determine the amount of processing power that is needed by a program.

# Types of Algorithms

The most common type of algorithms:

* Searching algorithms
* Pathfinding algorithm
* Sorting algorithms
* Compression algorithms
* Tree and graph-based algorithms
* Pattern matching algorithm among many others

**Conclusion**:

Different algorithms play different roles in programming. You only need to define your problem then select the right algorithm to use.

Q2 : What are the two different methods to represents an algorithm?

# Representing an algorithm:

There are two main ways that algorithms can be represented –

* Psaeudocode
* Flowcharts.

## Pseudocode:

Most programs are developed using programming languages. These languages have specific syntax that must be used so that the program will run properly. **Pseudocode is not a programming language**, it is a simple way of describing a set of instructions that does not have to use specific syntax.

Writing in pseudocode is similar to writing in a programming language. Each step of the algorithm is written on a line of its own in sequence. whenever**,** instructions **written in uppercase are**, variables **in lowercase and messages in sentence case.**In pseudocode, **INPUT**asks a question. **OUTPUT** prints a message on screen.

## Example:

A simple program could be created to ask someone their name and age, and to make a comment based on these. This program represented in pseudocode would look like this:

OUTPUT 'What is your name?'

INPUT user inputs their name

STORE the user's input in the **name** variable

OUTPUT 'Hello' + name

OUTPUT 'How old are you?'

INPUT user inputs their age

STORE the user's input in the **age** variable

IF age >= 70 THEN

OUTPUT 'You are aged to perfection!'

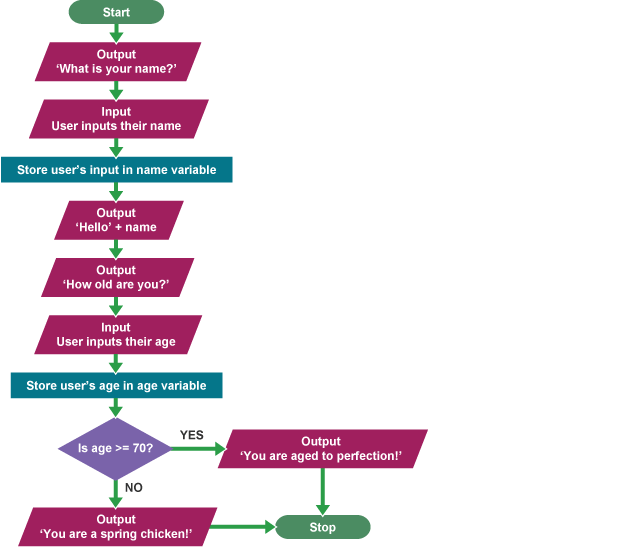
ELSE OUTPUT 'You are a spring chicken!'

# Representing an algorithm: Flowcharts

A flowchart is a diagram that represents a set of instructions. Flowcharts normally use standard symbols to represent the different instructions. There are few real rules about the level of detail needed in a flowchart. Sometimes flowcharts are broken down into many steps to provide a lot of detail about exactly what is happening. Sometimes they are simplified so that a number of steps occur in just one step.

## Example:

A simple program could be created to ask someone their name and age, and to make a comment based on these. This program represented as a flowchart would look like this:



Q3: Design a flowchart to find the average of two numbers?

start

Input x,y

Sum =average=0

Sum=x+y

Average=sum/2

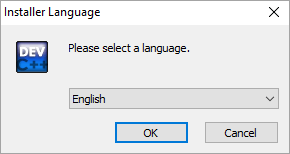
Print average

End

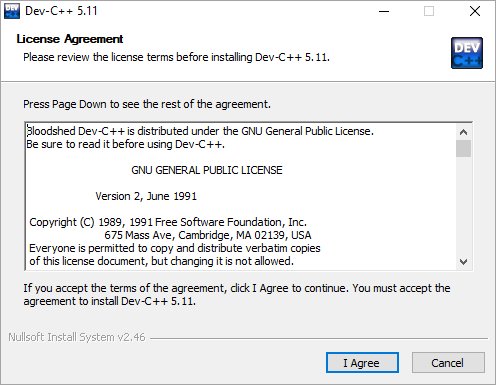
Q4: What are the steps for installation of Dev C++ (add screen shot for each step).

# The stepwise installation for dev-C++ is given below.

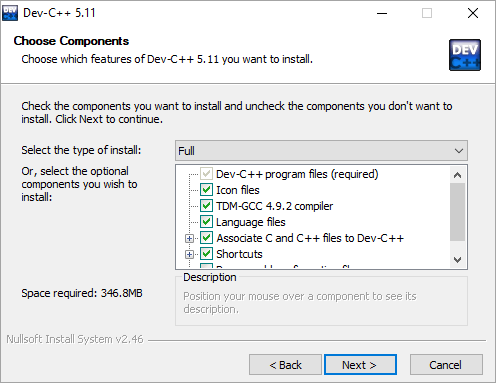
**#1)** The first step while we start the installer is to select the language of our choice as shown in the below screenshot.



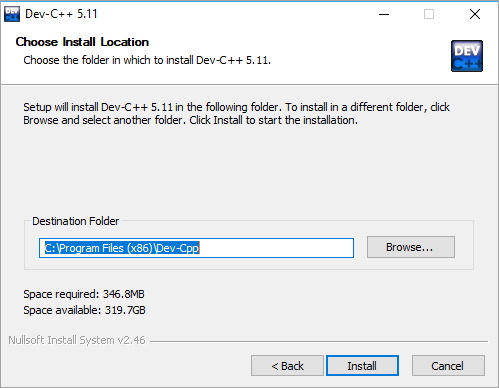
**#2)** Once you select the appropriate language, you have to agree to the license agreement that pop-ups next.



**#3)** Next, we are asked to select the components that we need to install as a part of the dev-C++ installation.

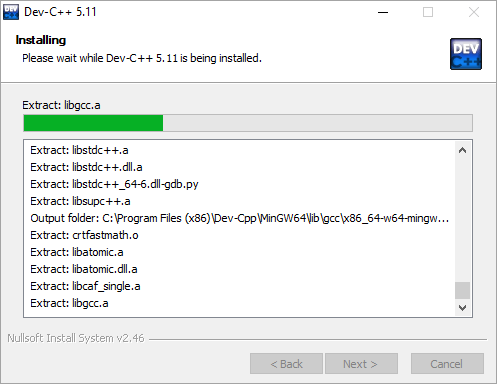


**#4)** Now the installer prompts the user for the destination folder where the dev-C++ files/libraries etc. are to be copied



Once we provide the destination folder path, click on Install

**#5)** The following screenshot shows the progress of the installation.

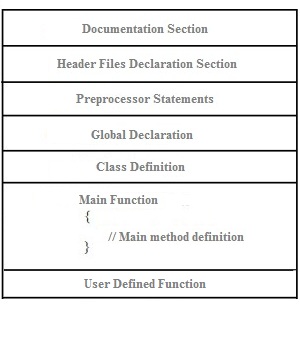


Once the installation is over, a “finish” dialog that signals the end of the installation appears. We click finish and then we can launch the dev-C++ IDE.

Q5: Explain structure of a C++ program?

# C++ Program Structure:

In C++, like any other programming language we have a specification available for program structure, every C++ program is consist of following building blocks



* **Documentation Section:** Documentation section is generally meant include set of comments, that is used to provide the information about the program written like name of a program, utility of program, date of creation, date of last modification ,author name, licensing or copyrights information and any other information that programmer wish to put for the references.

**Example:**

//Program Name: First C++ Program

/\*

* **Pre-processors Statements:** This is the section where we write all of the pre-processors directive statements, pre-processors statement usually with begins with a pound (#) symbol. Pre-processors statements are processed first when the compilation of the program begins. Pre-processors statements such **#include** tells compiler to include header files and **#define** directive is used to define constants prior to compilation of the program.

**Example:**

#include <iostream>

* **Global Declarations:** This is the section where all the global declaration comes. All of the variables, structures, classed and function defined or declared outside the main function are treated as global.
* **Class Definition:** The classes are used to map real world entities into programming. The Classes are the key building block of any C++ program. A C++ program may include several class definitions. This is the section where we define all of our classes.
* **Main Method Definition:** This is the most vital part of each and every C++ program, it is mandatory for C++ program to have a main() method definition. There can be only one main() method in C++ program. The execution of a C++ program starts with the main() method. The C++ program can not be executed without the main() method. The main() method is responsible for the execution of all the user defined statement, functions and library functions. The main() method further structured into – variable declaration, function declaration and user defined executable statements.
* **User defined functions:** This is the section of where we put all the user defined functions created to perform a specific task. A user defined function must be defined before use it. User defined function can written before or immediately after the main ( ) function and called inside the main ( ) function.