

جامعة
القاهرة الجديدة
التكنولوجية



NEW CAIRO
TECHNOLOGICAL
UNIVERSITY





TUE – The Technological Universities in Egypt
NCTU – New Cairo Technological University
Faculty of Industry and Energy Technology
Information Technology Department
First-Year

Course: Programming Essentials in C

Lecture 5

Presented by

Dr. Ghada Maher

Contents

Difference between an Assembler and a Compiler

Difference between Syntax and Semantics

Difference between Algorithm, Pseudocode , flowchart.

Switch Case Control Statement

Switch Case control statements allow the computer to take a decision as to which statement is to be executed next.

```
switch(expression)
{
case 1:  statement 1 sequence;
        break;
case 2:  statement 2 sequence;
        break;
--
--
--
case n:  statement n sequence;
        break;
default : default statement sequence;
}
```

Example: write a C program to create a calculator which can do addition, subtraction, multiplication, division, & modulus (remainder) using Switch Case control statements

```
1  #include <stdio.h>
2  int main()
3  {
4      int a,b,ch;
5      printf("Enter two nos. ");
6      scanf("%d%d",&a,&b);
7      printf("1:\tAdd\n2:\tSubtract\n3:\tMultiply\n4:\tDivide\n:\tModulus\nEnter
      your choice: ");
8      scanf("%d",&ch);
9      switch(ch)
10     {
11     case 1:
12     printf("%d\n",a+b);
13     break;
14     case 2:
15     printf("%d\n",a-b);
16     break;
17     case 3:
18     printf("%d\n",a*b);
19     break;
20     case 4:
21     printf("%f\n",(float)a/b);
22     break;
23     case 5:
24     printf("%d\n",a%b);
25     break;
26     default:
27     printf("Invalid Input");
28     }
29     return 0;
30 }
```

Assembler, Compiler, and interpreter

In general, compiler is a computer program that reads a program written in one language, which is called the source language, and translates it in to another language, which is called the target language. Traditionally, source language was a high level language such as C++ and target language was a low level language such as Assembly language.

However, there are compilers that can convert a source program written in Assembly language and convert it to machine code or object code. Assemblers are such tools. So, both assemblers and compilers ultimately produce code that can be directly executed on a machine.

A interpreter is similar to sentence-by-sentence translation, whereas a compiler is similar to translation to the whole passage.

Difference between: Assembler and Compiler.

Assembler	Compiler
It translates the mnemonic codes such as PRN, ADD, and SUB etc. To machine language code.	It translates the high level language to assembly language.
The program, which executes using assembler, executes faster, because it directly converts the source code in machine language.	It takes time to execute a program, because it first translates the source code into another compiler's language and then using assembler converts it into machine language.

Difference between: Compiler and Interpreter

Compiler	Interpreter
Compiler translates the entire high level language program into the machine language program at once before executing it. This optimizes the use of machine language instructions in the translated program. Therefore normally compiled programs run faster than Interpreted programs. The original high level language program is called a <i>source program</i> . The compiled program i.e. machine language program generated by the compiler after translation is called object program.	The Interpreter translates the program written in high level language into machine language at the time of executing that program, instructions by instructions. That is, it reads the first instruction written in the program and converts that into equivalent machine language instructions. Then the CPU watches those machine language instructions. After that, the Interpreter reads and translates the next instruction and so on.
Compiler Takes Entire program as input	Interpreter Takes Single instruction as input.
Object code is permanently saved for future use.	No object code is saved for future use.

Compiler	Interpreter
Non time consuming translation method.	Time consuming translation method
Non time consuming translation method.	Time consuming translation method
It requires large space in the computer.	Interpreter are easy to write and do not require large memory space.
Speed of a compiler is very fast.	Speed is very slow.
Any change in source program after the compilation requires recompiling of entire code.	Any change in source program during the translation does not require's retranslation of entire code.
Intermediate Object Code is Generated .	No Intermediate Object Code is Generated .
Conditional Control Statements are Executes faster .	Conditional Control Statements are Executes slower .
Memory Requirement : More (Since Object Code is Generated)	Memory Requirement is Less.
Program need not be compiled every time.	Every time higher level program is converted into lower level program.
Errors are displayed after entire program is checked.	Errors are displayed for every instruction interpreted (if any)
Example : C Compiler.	Example : BASIC.

Difference between Syntax and Semantics

Semantics	
<p>It is the logic or planning of the program. Semantics can be written in any of the following ways:</p> <ol style="list-style-type: none">1. Flowcharts.2. Algorithms.3. Pseudo codes.	<p>It is the way of writing the program in a particular programming language. Syntax changes from language to language.</p>

FLOWCHART

Symbols:-



Start / Stop Box (Ellipse)



Input or Output Box (Parallelogram)



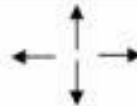
Process or Calculation Box (Rectangle)



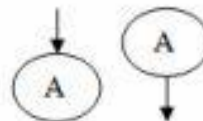
Condition or Decision Box



Function Call or Sub-Program Call Box



Control Flow Lines (Path or direction)



Connector (To join to different parts of a single program)

ALGORITHMS

- Once a problem is been properly defined, a detailed, finite, step-by-step procedure for solving it must be developed. This procedure is known as algorithm.
- Algorithm can be written in ordinary language, or using formal procedures that lie somewhere between ordinary and programming languages.

PSEUDOCODE

- Sometimes, it is desirable to translate an algorithm to an intermediate form, between that of a flowchart and the source code. Pseudocode is an English approximation of source code that follows the rules, style, and format of a language but ignores most punctuation's.

Write an Algorithm, Pseudocode , and flowchart to add two numbers.

Algorithm to add two numbers.

Step 1: Start

Step 2: Declare variables num1, num2 and sum.

Step 3: Read values for num1, num2.

Step 4: Add num1 and num2 and assign the result to a variable sum.

Step 5: Display sum

Step 6: Stop

Pseudocode to add two numbers

- integer num1, num2, and sum;
- read in num1 and num2;
- add num1 & num2 and set it to sum;
- write sum;

Flowchart to add two numbers.

