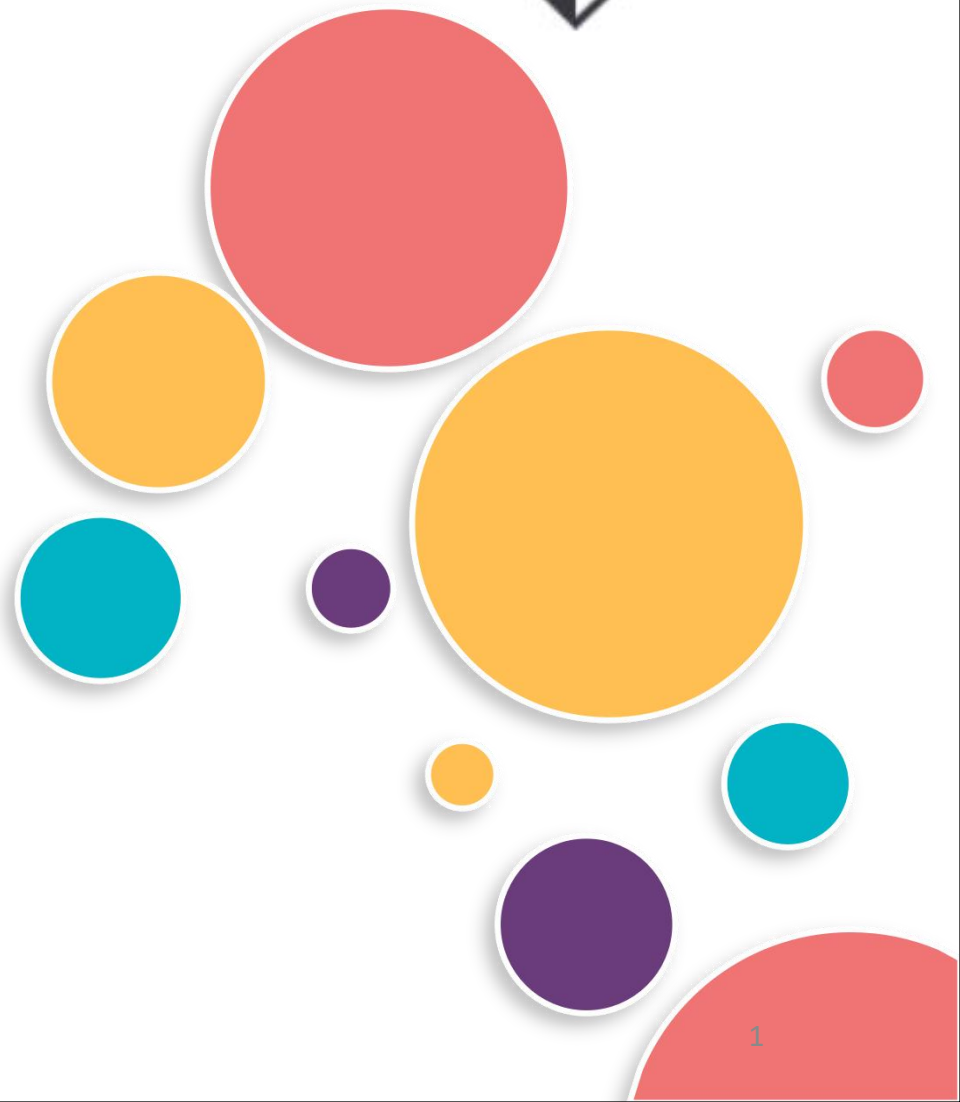
SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY

ECT 121
Computer Programming I

Dr. Amina Elhawary



Course Assessment

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

- **The course assessment comprises the following components:**

➢ Final Exam	(30%)
➢ Midterm Exam	(20%)
➢ Tutorial Work (Quizzes, Assignments,..)	(20%)
➢ Lab Work (Project, Assignments,..)	(20%)
➢ Final Course Project	(10%)

References

- C program Design for Engineers, Addison wesley, Jer R. Hanly, 2nd edition, 2000.

- An Introduction to the C Programming Language and Software Design, Tim Bailey, 2005.

SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY

Course Outlines

- An Introduction to Computer Programming
- Data Types, Operators and Simple functions
- Selection Structure (if and Switch statement)
- Loop and Nested Loop statements
- Arrays and Applications
- Pointers
- Function and Modulator programming
- Structures

SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY

Lecture 1

An Introduction to Computer Programming

What is Computer Programming ?

SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY

- Computers process data under the control of sets of instructions called computer programs.
- These programs guide the computer through ordered actions specified by people called computer programmers.
- The programs that run on a computer are referred to as a software.

What is Programming?

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

- *Given a well-defined problem:*

1. Find an algorithm to solve a problem.

2. Express that algorithm in a way that the computer can execute it.

What is Programming Language ?

SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY

- ***Programming Languages***

Programmers write instructions in various programming languages, some directly understandable
by computers and others requiring intermediate *translation* steps.

What is Programming Language ?

SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY

1. Machine Languages

Any computer can directly understand only its own machine language, defined by its hardware architecture. Machine languages generally consist of numbers (ultimately reduced to 1s and 0s). Such languages are cumbersome for humans.

Ex:

```
10100001 10111100 10010011 00000100
00001000 00000011 00000101 11000000
10010011 00000100 00001000 10100011
11000000 10010100 00000100 00001000
```

It means: `z = x + y;`

What is Programming Language ?

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

2. Assembly Languages

Programming in machine language was simply too slow and tedious for most programmers. Instead, they began using English like abbreviations to represent elementary operations. These abbreviations formed the basis of assembly languages. Translator programs called assemblers were developed to convert assembly-language programs to machine language. Although assembly-language code is clearer to humans, it's incomprehensible to computers until translated to machine language.

Ex:

D = A * B + 10

Intel Assembly Language:

```
mov   eax, A
mul   B
add   eax, 10
mov   D, eax
```

What is Register ?

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

A **register** is one of a small set of data holding places that are part of a computer processor . A **register** may hold :

- a computer instruction
- a storage address
- or any kind of data (such as a bit sequence or individual characters)

What is Programming Language ?

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

3. High Level Languages

- To speed the programming process even further, high-level languages were developed in which single statements could be written to accomplish substantial tasks.
- High-level languages allow you to write instructions that look almost like everyday English and contain commonly used mathematical expressions.

- **Compilers**

Translator programs called compilers convert high-level language programs into machine language. The process of compiling a large high-level language program into machine language can take a considerable amount of computer time.

Software Development Life Cycle (SDLC)

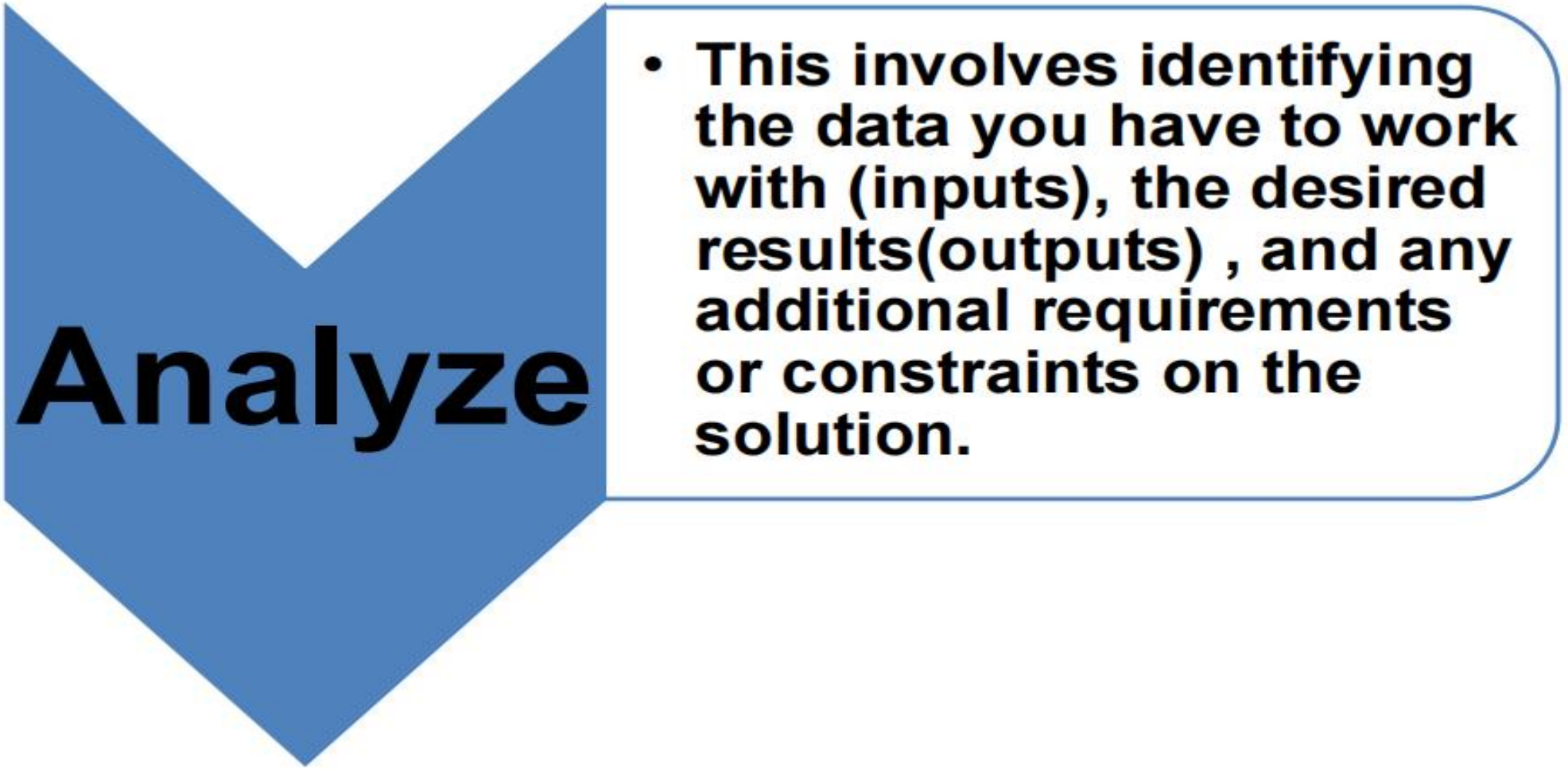
SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

- SDLC = The development of a system from the time it is first studied until the time it is updated or replaced

Software Development Life Cycle

Software Development Life Cycle (SDLC)

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY



Software Development Life Cycle

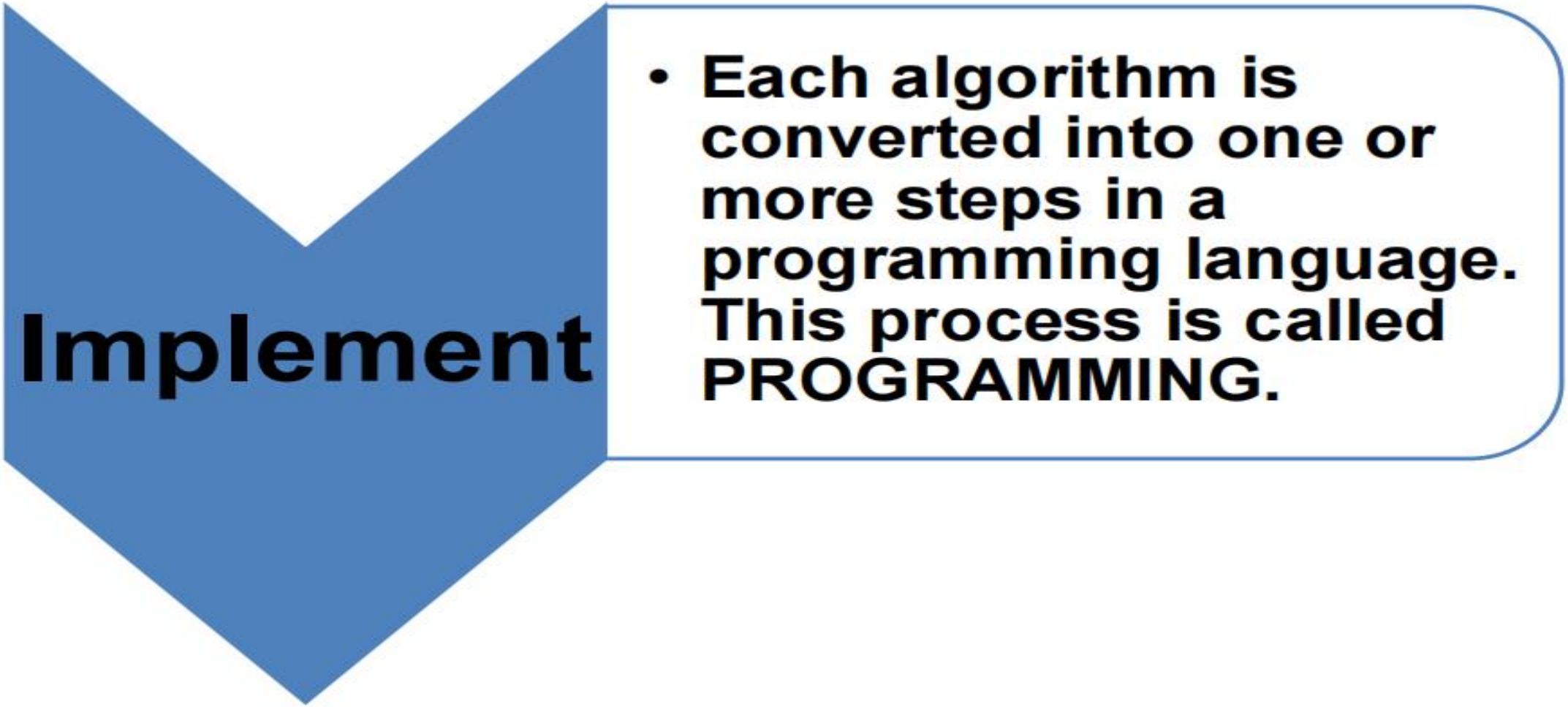
SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY

Design

- **An algorithm is a step-by-step procedure for solving a problem in a finite amount of time.**

- **Algorithm for a programming problem**
- 1-Get the data
- 2-Perform the computation
- 3-Display the results.

Software Development Life Cycle (SDLC) Cont.



Software Development Life Cycle (SDLC) Cont

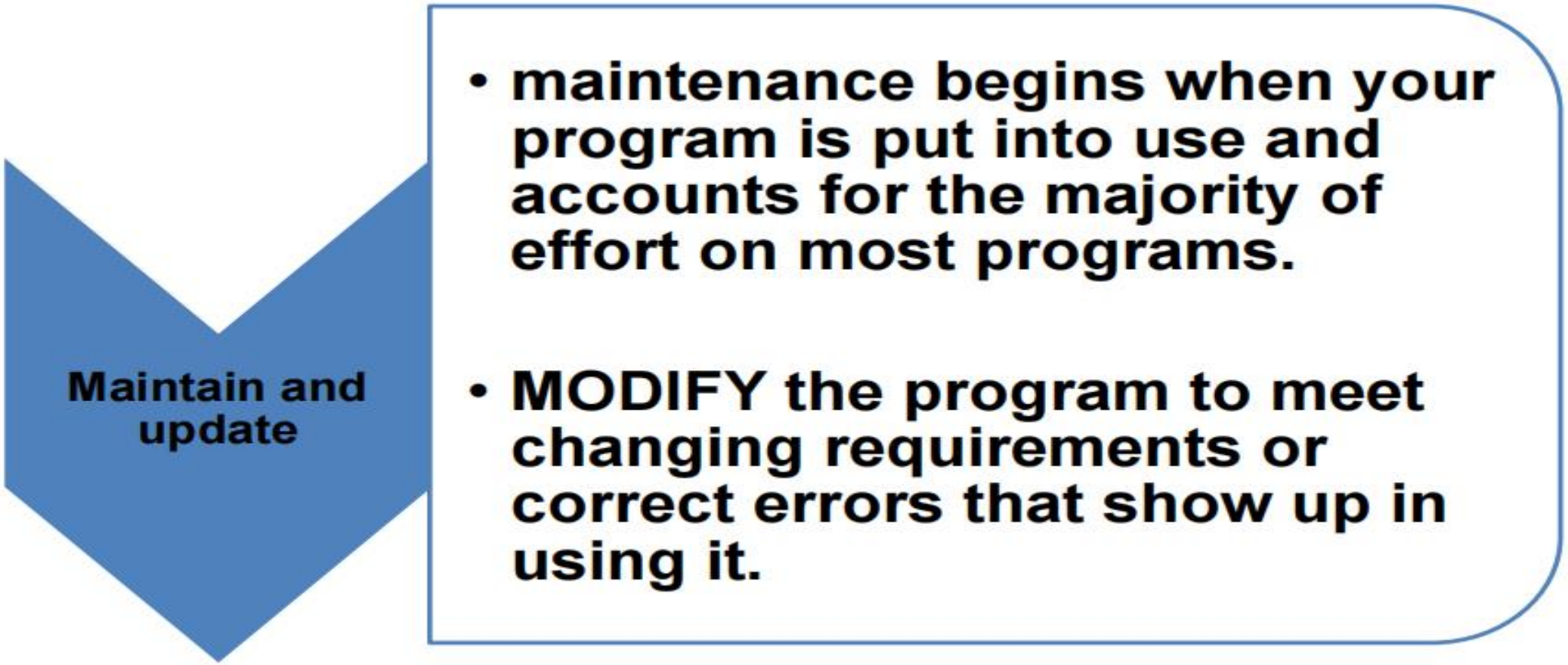
SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY

Test and verify

- Run the program several times using different sets of data, making sure that it works correctly for every situation in the algorithm .
- if it does not work correctly, then you must find out what is wrong with your program or algorithm and fix it--this is called DEBUGGING.

Software Development Life Cycle (SDLC) Cont.

SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY



The C Programming Language

SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY

- ## The C Standard Library

C programs consist of pieces called functions. You can program all the functions that you need to form a C program, but most C programmers take advantage of the rich collection of existing functions called the C Standard Library. Thus, there are really two parts to learning how to program in C learning the C language itself and learning how to use the functions in the C Standard Library.

The C Programming Language (Cont.)

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

```
1  // Fig. 2.1: fig02_01.c
2  // A first program in C.
3  #include <stdio.h>
4
5  // function main begins program execution
6  int main( void )
7  {
8     printf( "Welcome to C!\n" );
9  } // end function main
```

```
Welcome to C!
```

Comments

- Even though this program is simple, it illustrates several important features of the C language. Lines 1 and 2 begin with //, indicating that these two lines are comments.
- You insert comments to document programs and improve program readability. Comments do *not* cause the computer to perform any action when the program is run.
- Comments are *ignored* by the C compiler and do *not* cause any machine-language object code to be generated. The preceding comment simply describes the figure number, file name and purpose of the program.
- Comments also help other people read and understand your program.

The C Programming Language (Cont.)

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

include Preprocessor director

```
1  // Fig. 2.1: fig02_01.c
2  // A first program in C.
3  #include <stdio.h>
4
```

- **#include** <stdio.h> is a directive to the **C preprocessor**.
- Lines beginning with # are processed by the preprocessor *before* compilation.
- Line 3 tells the preprocessor to include the contents of the **standard input/output header** (**<stdio.h>**) in the program. This header contains information used by the compiler when compiling calls to standard input/output library functions such as printf (line 8).

The C Programming Language (Cont.)

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

```c
// Fig. 2.1: fig02_01.c
// A first program in C.
#include <stdio.h>

// function main begins program execution
int main( void )
{
   printf( "Welcome to C!\n" );
} // end function main
```

```
Welcome to C!
```

1 Input Functions (Take User Input)

Function	Description	Example
`scanf()`	Reads formatted input	`scanf("%d", &num);`
`getchar()`	Reads a single character	`char ch = getchar();`
`gets()`	Reads a string (deprecated)	`gets(str);`
`fscanf()`	Reads formatted input from a file	`fscanf(fp, "%d", &num);`

2 Output Functions (Display Output)

Function	Description	Example
`printf()`	Prints formatted output	`printf("Hello, World!\n");`
`putchar()`	Prints a single character	`putchar('A');`
`puts()`	Prints a string	`puts("Hello");`
`fprintf()`	Writes formatted output to a file	`fprintf(fp, "Data: %d", num);`

The C Programming Language (Cont.)

SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY

The main Function

```
5  // function main begins program execution
6  int main( void )
```

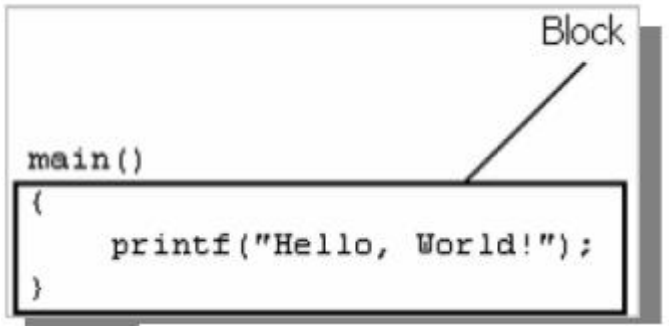
- Line 6 is a part of every C program.
- The parentheses after main indicate that main is a program building block called a function. C programs contain one or more functions, one of which *must* be main.
- Every program in C begins executing at the function main.

The C Programming Language (Cont.)

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

```c
#include<stdio.h>

main()
{
	printf("Hello, World!");
}
```




```c
#include<stdio.h>

main()
{
	printf("Hello, "), printf("World!");
}
```

```c
#include<stdio.h>

main()
{
	printf("Hello, ");
	printf("World!");
}
```

```c
#include<stdio.h>

main()
{
	printf("Hello, "); printf("World!");
}
```

```c
#include<stdio.h>

main()
{
```

```c
	printf("Hello, \nWorld!");
}
```

How to change c code to c++ code

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

C code

```c
// Fig. 2.1: fig02_01.c
// A first program in C.
#include <stdio.h>

// function main begins program execution
int main( void )
{
   printf( "Welcome to C!\n" );
} // end function main
```

```
Welcome to C!
```

C++ code

```cpp
#include <iostream>


int main() {

    std::cout << "Hello, World!" << std::endl;

    return 0;

}
```

```cpp
#include <iostream>

int main() {
    int number;

    std::cout << "Enter a number: ";
    std::cin >> number;

    std::cout << "You entered: " << number << std::endl;

    return 0;
}
```

Input

```
Enter a number: 5
```

Output

```
You entered: 5
```

How to change c code to c++ code

SAXONY EGYPT UNIVERSITY FOR APPLIED SCIENCE AND TECHNOLOGY

Another way

```
#include <iostream>
using namespace std;

int main() {
    cout << "Welcome to C++!" << endl;
    return 0;
}
```

- When writing C++ programs, many standard functions (like cout, cin, endl, vector, string, etc.) belong to the std namespace. Instead of writing std::cout, std::cin, and so on every time.

- When writing C++ programs, many standard functions (like cout, cin, endl, vector, string, etc.) belong to the std namespace. Instead of writing std::cout, std::cin, and so on every time.

SAXONY EGYPT UNIVERSITY
FOR APPLIED SCIENCE AND TECHNOLOGY

THANK YOU