

Name: Muhammad tayyab

ID # 7945

Sec " B "

Sub: Introduction to Computer
programming.

Instructor: Engr: Syed Ashraf
Ali

Department of Civil
Engineering

IQRA NATIONAL UNIVERSITY

PESHAWER

Q No#1: Design an algorithm and draw a flow chart that will read the two side of a rectangle and calculate its area.?

ANS:

• Pseudo Code:

- Input the width (W) and length (L) of a rectangle,
- Calculate the Area (A) by multiplying L with W,
- ✓ Print A

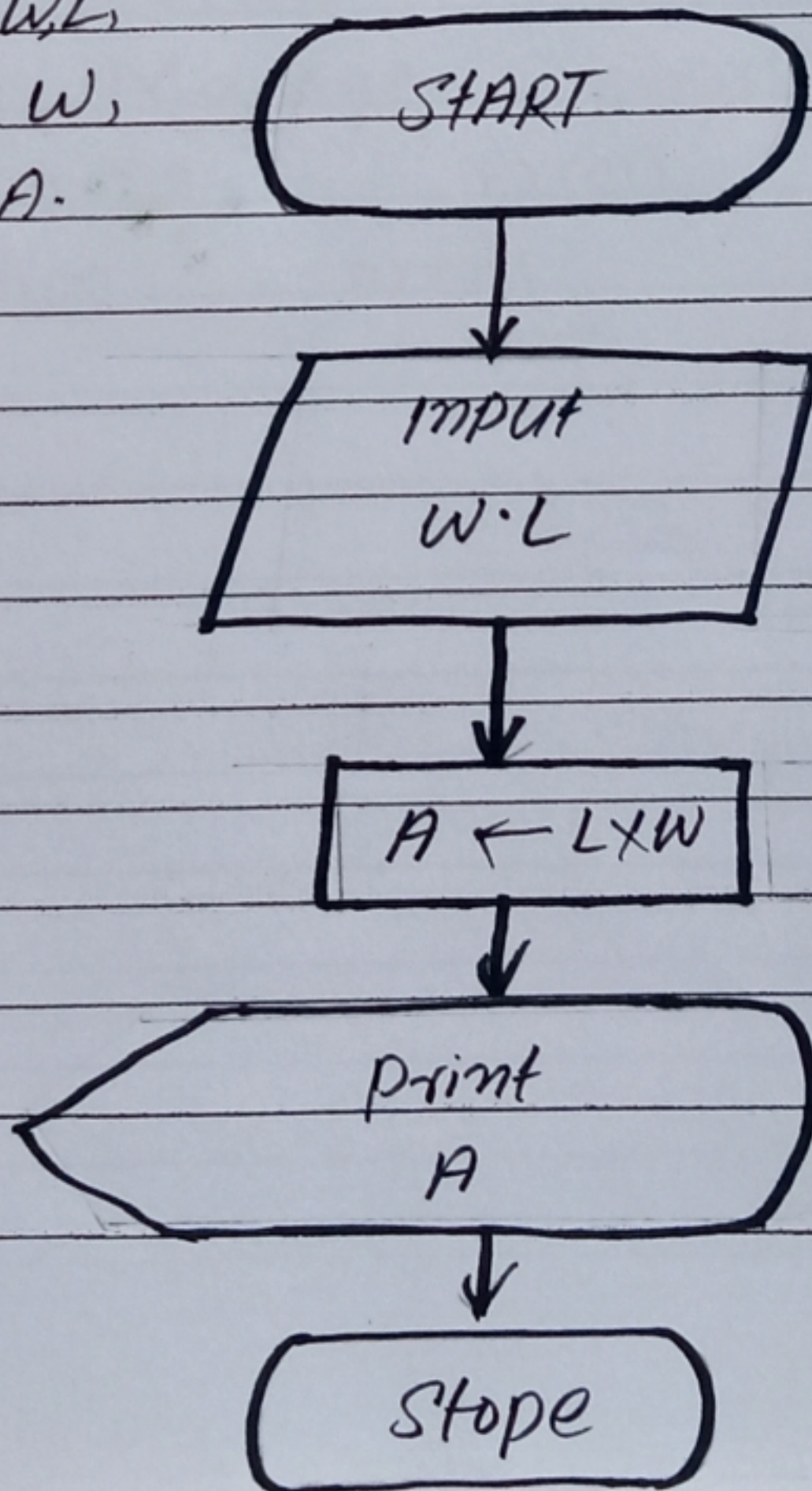
Algorithms & Flowcharts.

• Algorithm;

Step 1: Input W, L

Step 2: $A \leftarrow L \times W$,

Step 3: Print A.



Q No # 01 part (b)

Name different types of errors which can occur during the execution of a program.

Ans: Following are the Name of the different types of errors which can occur during the execution of a program.

* Syntax errors

* system errors

* Runtime errors

* Stop errors

* Logical errors

* Device Manager errors

* Post Code errors

* Application errors.

*

Q: No 2: Part (a): Why we use
iostream.h and conio.h in C++
Programming?

Ans: <iostream.h>

* This is the name of the library definition file for all input output streams. Your program will almost certainly want to send stuff to the screen and read things from the keyboard.

* iostream.h is the name of the file in which has code to do that work for you.

<conio.h>

• conio.h header file used in programming language for console input and output.

• Some of the most commonly used functions of conio.h are

(1) clrscr ()

(2) getch ()

Q No # 02 part (b):

Program maintenance is the process of modifying a software OR program after delivery to achieve any of these outcomes.

- * Correct errors.

- * Improve performance.

- * Add functionalities.

- * Remove obsolete portions.

Despite the common perception that maintenance is required to fix errors that come up after the software goes live in reality most of the maintenance work involves adding minor OR major capabilities to existing modules. For example, some new data is added to a report, a new field added to entry forms, code to be modified to incorporate changed government laws etc.

Types of Maintenance

- * Corrective Maintenance

* Preventive maintenance.

* Adaptive maintenance

* Perfective maintenance

∴ Maintenance tools:-

* Program slicer :-> Selects a part of the program that would be affected by the ~~rectangle~~ change.

* Data flow analyzer: tracks all possible flows of data in the software.

* Dynamic analyzer: traces program execution path

* Dependency analyzer:-> assists in understanding and analyzing independence of different parts of the program.

* Static analyzer:- allow general viewing and summarizing of the programs.

Now what is update of
a program.

When we perform an update it involves making changes to an app or an operating system is such a way that it doesn't affect its core structure. So, most of the frequent changes made to your computer like bug fixes, security patches, adding support for drivers and newer hardware etc. can be termed as an update.

An update is often small in size and it might take a couple of minutes to perform one. The word "small" is relative. For instance an update meant for a single app can range from few kilobytes to a couple of megabytes in size, while an update for an operating system can go up to a few hundred megabytes.

Q no # 03

Part (a) ANS.

-: Bug and Debug :-

A Bug is a coding error in a computer program. The process of finding bugs before program users do is called debugging. Debugging starts after the code is first written continues in successive stages as code is combined with other units of programming to form a software product such as Operating System OR an application. After a product is released or during public data testing, bugs are still apt to be discovered. When this occurs, users have to either find a way to avoid using the "buggy" code OR get a patch from the originators of the code.

Although bugs typically just cause annoying computer glitches, their impact can be much more serious.

Q No # 03

Part (b) ANS.

Syntax error and logical error.

A syntax error is an error in the source code of a program. Since computer programs must follow strict syntax to compile correctly, any aspects of the code that do not conform to the syntax of the programming language will produce a syntax error.

While the logical error is a logic error (OR Logical error) is a mistake in a program's source code that results in incorrect or unexpected behavior.

It is a type of runtime error that may simply produce the wrong output or may cause a program to crash while running.

Many different types of error or programming mistakes can cause logic errors. For example, assuming a value to be the wrong variable may cause a series of unexpected program errors.

Compiler:

• A compiler is a program that translates user readable source code into computer executable machine code.

◦ To do this successfully the human readable code must comply with the syntax rules of whichever programming language it is written in.

◦ The compiler is only a program and cannot fix your programs for you. If you make a mistake, you have to correct the syntax or it won't compile.

◦ What happens when you compile code; a compile complexity depends on the syntax of the language and how much abstraction that programming language provides.

◦ compiling:

◦ To write a program takes these steps:

• Edit the program,

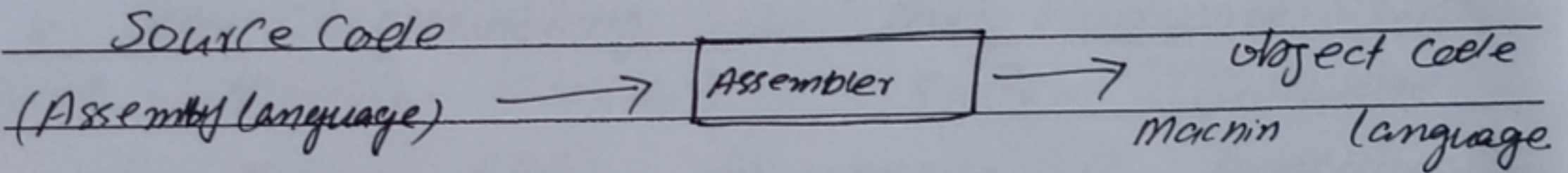
• Compile the program into machine code files,

• Link the machine code files into a able program,

• Run the program with some languages.

Q Assembler:

The Assembler is used to translate the program written in Assembly language into machine code. The source program is an input of assembler that contains assembly language instructions. The output generated by assembler is the object code or machine code understandable by the computer.



Part (d) : System Software & Application Software?

Ans: System Software :

- * Software which is used to perform the basic functionality of the computer or in other words bridge between used & hardware.
- * Also know as operating system.
- * After assembling the hardware first of all we install system software on the (PC) to make it functional.
- * It is an necessary part in the operating of the computer.
- * We cant make personal computer functional without the system software.
- * Is collection of programs designed to operate, control and extend the processing capabilities of the computer itself.
- * Usually developed by computer manufacturers in a low level language.
- * Makes operation of computer more effective and efficient.
- * Help hardware components work together and provide support for the

for the development and execution of the application software.

* Programs included in system software package are known as system programs and the programmers who develop those programs are known as system programmer.

* Examples are:

- x operating systems.
- x utilities programs,
- * communication software,
- * assembler, interpreter etc.

✓ Types of System software:

- * operating system,
- * utilities (service programs),
- * device drivers,
- * language translators.

Application Software:

- Those Software which used to perform the specific tasks of the computer are known as application software.
- This software perform only a single task.
- They comes in different categories:
 - Anti Viruses,
 - Norton Anti viruses,
 - Text Editing,
 - Notepad, wordpad, MS Word.
 - Presentation Making,
 - MS Power Point.
 - Calculation Software.
 - MS Excel.
 - Compress Software,
 - Winzip, WinRAR.

* Feature of application software.

- Consist of programs designed to make users more productive and/or assist with personal makes.
 - Helps users solve particular problems,
 - In most cases, application software resides on the computer's hard disk,
 - Application software can also be stored on CDs, DVDs and flash or keychain storage devices.

6

* Categories of Application Software:

- Business Software,
- Graphic & Multimedia
- Home / Personal / Education,
- Communication

(e) Low Level language:

- Computer language consisting of mnemonics that directly correspond to machine language instructions.

* Low level languages.

- very close to machine language.
- Concentrate on machine architecture.

• High Level Language:

- Basically symbolic languages that used English words and or mathematically symbols rather than mnemonic codes.
- machine-independent programming language.
- Concentrate on the logic of problem.
- High level language.
 - C,
 - C++
 - Java,

Low Vs. High Level language:

- Differences in terms of:
 - Understand,
 - Ease of Writing.
 - Running speed,
 - Writing format,

⇒ Understandable:

- Low Level Language
- Mnemonic, binary, hexadecimal.
- Adds two numbers and stores the result

model Small, C

.586 .

.data

move eax, 5

move ebx, 10

add eax ebx

end

Adds two numbers and stores the result.

Ease of Writing.

Low Level Language

Designed for the ease of the computer running the language.

Difficult for users to read and write.

↳ Running speed: 7

- Faster
- no need to compile
- more efficient.

Understandable:

- High Level Language.

Simple English and mathematics symbols.

- Add two numbers and stores the result.

High Level Language

```
int main ()  
{
```

```
    // assign to the variable result the  
    value of 5+10 int result = 5+10;  
    return 0;  
}
```

• Ease of Writing:

- High level language.

Designed for the ease of the person writing the language.

- using language that user can understand English.

Running Speed:

High level language.

Need compilers or interpreter:

- translate into machine code.

lower speed execution.