

## Mid term Assignment/Quiz



Submitted by	Wajid ali
ID	7636
Subject	C++(theory)
Program	BS(Civil)
Semester	10 <sup>th</sup>
Date	13/04/2020

IQRA NATIONAL UNIVERSITY  
PESHAWAR

**Q.1 (a) Design an algorithm and draw a flowchart that will read the two sides of a rectangle and calculate its area.**

**(b) Name different types of errors which can occur during the execution of a program.**

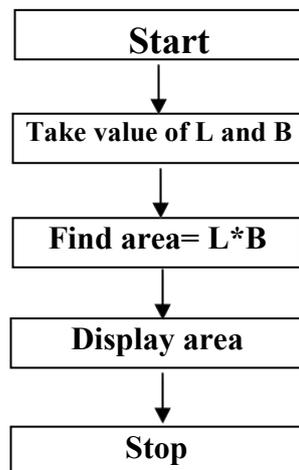
**ANS**

**Part (a)**

### **Algorithm**

1. Start
2. Take value of L and B
3. Find area= L\*B
4. Display area
5. Stop

### **Flowchart**



**Part (b)**

There are three kinds of errors:

#### **1) Syntax errors:**

Errors occur when our program contains grammatical errors. A syntax error is an error in the source code of a program. ... For example, a missing semicolon at the end of a line or an extra bracket at the end of a function may produce a syntax error.

## 2) Logical errors:

Errors such as calculation mistakes etc. Logic errors occur when a program does not do what the programmer expects it to do

## 3) Run time errors:

These errors occur while the program is running. A run time error is an application error that occurs during program execution. Run time errors are usually a category of exception that

encompasses a variety of more specific error types such as logic errors, IO errors, encoding errors, undefined object errors, division by zero errors

**Q.2 (a) Design an algorithm that reads two values, determines the largest value and prints the largest value with an identifying message.**

**(b) What do you understand by the term “Maintain and update the Program”.**

### Ans Part (a)

#### Algorithm

Step 1: Start

Step 1: Input VALUE1, VALUE2

Step 2: if (VALUE1 > VALUE2) then MAX ← VALUE1 else MAX ← VALUE2 end if

Step 3: Print “The largest value is”, MAX

### Part (b)

Maintain and Update the program. Maintenance and update are the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment

- ❖ **Software maintenance** is a part of **Software Development Life Cycle**. Its main purpose is to modify and **update software** application after delivery to correct faults and to improve performance. When the real world changes,

the **software** requires alteration wherever possible.

- ❖ An **update** is new, improved, or fixed software, which replaces older versions of the same software. For example, **updating** your operating system brings it up-to-date with the latest drivers, system utilities, and security software. **Updates** are often provided by the software publisher free of additional charge.

**Q.3 Differentiate between the following.**

- (a) **Bug & Debug**
- (b) **Syntax error & Logical error**
- (c) **Compiler & Assembler**
- (d) **System Software & Application Software**
- (e) **Low level language & High level language**

<b>Bug</b>	<b>Debug</b>
<p>A software bug is an</p> <ul style="list-style-type: none"> <li>• Error,</li> <li>• Flaw,</li> <li>• Failure,</li> <li>• (or) Fault</li> </ul> <ul style="list-style-type: none"> <li>➤ In a computer program or system that produces an incorrect or unexpected result, or causes it to behave in unintended ways.</li> <li>➤ Most bugs arise from mistakes and errors made by people in either a program's source code or its design, and a few are caused by compilers producing incorrect code.</li> <li>➤ A program that contains a large number of bugs, and/or bugs that seriously interfere with its functionality, is said to be buggy. Reports detailing bugs in a program are commonly known as bug reports, defect reports, fault reports, problem reports, trouble reports, change requests, and so forth.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Debugging is the process of locating and fixing errors (known as bugs), in a computer program, or hardware device.</li> <li>➤ Debugging is a necessary process in almost any new software, or hardware development process, whether a commercial product, an enterprise, or personal application program.</li> <li>➤ For complex products, debugging is done periodically throughout the development, and again during the customer beta test stages. Because most computer programs and many programmed hardware devices contain thousands of lines of code, almost any new product is likely to contain a few bugs. Invariably, the bugs in the functions that get the most use, are found and fixed first.</li> </ul>

## Syntax error & Logical error

<b>Syntax error</b>	<b>Logical error</b>
<ul style="list-style-type: none"><li>❖ A syntax error is an error in the syntax of a sequence of characters or tokens that is intended to be written in particular programming language.</li><li>1. A syntax error occurs due to fault in the program syntax.</li><li>2. In compiled languages the compiler indicates the syntax error with the location and what the error is.</li><li>3. It is easier to identify a syntax error.</li></ul>	<ul style="list-style-type: none"><li>❖ Logical error is an error in a program that causes it to operate incorrectly but not to terminate abnormally.</li><li>1. A logical error occurs due to a fault in the algorithm.</li><li>2. The programmer has to detect the error by himself.</li><li>3. It is comparatively difficult to identify a logical error.</li></ul>

## Compiler & Assembler

<b>Compiler</b>	<b>Assembler</b>
<ul style="list-style-type: none"><li>❖ The compiler is a computer program that takes the source code and translates it into the assembly code.</li><li>1. Compiler input source code.</li><li>2. Phases of the compiler are a lexical analyzer, syntax analyzer, semantic analyzer, intermediate code generated, a code optimizer, code generator, symbol table, and error handler.</li><li>3. The output of the compiler is a mnemonic version of the code.</li></ul>	<ul style="list-style-type: none"><li>❖ Assembler is a computer program that takes the assembly code generated by the compiler and translates it into the machine code.</li><li>1. Assembler input assembly language code.</li><li>2. Phases of assembler are the first phase and the second phase.</li><li>3. The output of assembler binary code.</li></ul>

## System software and Application software.

<b>System software</b>	<b>Application software</b>
<ul style="list-style-type: none"><li>➤ System software is used for operating computer hardware.</li><li>➤ System software are installed on the computer when operating system is installed.</li><li>➤ In general, the user does not interact with system software because it works in the background.</li><li>➤ System software can run independently. It provides platform for running application software.</li></ul>	<ul style="list-style-type: none"><li>➤ Application software is used by user to perform specific task.</li><li>➤ Application software are installed according to user's requirements.</li><li>➤ In general, the user interacts with application software.</li><li>➤ Application software can't run independently. They can't run without the presence of system software.</li><li>➤ Some examples of application software are word processor, web browser, media player etc.</li></ul>

## Low level language & High level language

<b>Low level language</b>	<b>High level language</b>
<ol style="list-style-type: none"><li>1. Low level language is the challenging to learn and understand.</li><li>2. They execute with high speed.</li><li>3. They allow little or no abstraction.</li><li>4. Modifying programs is difficult.</li><li>5. It is machine friendly language.</li><li>6. Low level language is high memory efficient.</li><li>7. It is tough to understand.</li><li>8. It is complex to debug comparatively.</li></ol>	<ol style="list-style-type: none"><li>1. High level language is easy to learn and understand.</li><li>2. They are executed slower than lower level language because they require a translator program.</li><li>3. They allow much more abstraction.</li><li>4. The programs are easy to modify.</li><li>5. It is programmer friendly language.</li><li>6. High level language is less memory efficient.</li><li>7. It is easy to understand.</li></ol>

<p>9. It is complex to maintain comparatively.</p> <p>10. It is non-portable.</p> <p>11. It is machine- dependent.</p> <p>12. It needs assembler for translation.</p> <p>13. Machine language and Assembly language are low-level languages.</p>	<p>8. It is simple to debug.</p> <p>9. It is simple to maintain.</p> <p>10. It is portable.</p> <p>11. It can run on any platform.</p> <p>12. It needs compiler or interpreter for translation.</p> <p>BASIC, Perl, Pascal, COBOL, Ruby etc are example of high level languages.</p>
--	--