***DEFINITIONS***

***Chapter 1***

A computer is an electronic device, operating under the control of instructions stored in its own memory.

INPUT–Hardware devices, such as keyboards and mice, perform input operations–Data, or facts, enter the computer system through input devices

PROCESSING–Processing data items may involve organizing them, checking them for accuracy, or performing mathematical operations on them‒The hardware that performs these tasks is the central processing unit, or CPU

OUTPUT–Sending information resulting from processing to a printer or monitor so people can view, interpret, and use the results–Can be stored on disks or flash media for later retrieval

Elements of a Computer System: **Hardware** - Central Processing Unit (CPU) / Main memory (primary storage): RAM / Secondary storage / Input/Output devices – **Software**

**CPU:** The processor or the central processing unit CPU )), interprets and carries out the basic instructions that operate a computer (Control Unit) and arithmetic and logical operations (Arithmetic Logic Unit) - Brain of the computer - One of the most expensive piece of hardware - Multiple Cores allows for parallel processing.

**Main Memory**

Random access memory (RAM): Directly connected to the CPU - All programs must be loaded into main memory before they can be executed - All data must be brought into main memory before it can be manipulated - When computer power is turned off, everything in main memory is lost

**Secondary storage:** device that stores information permanently / Examples: – Hard disks Flash drives Floppy disks Zip disks CD ROMs Tapes

**Input devices:** feed data and programs into computers (Keyboard, mouse, secondary storage)

**Output devices:** display results (Monitor, Printer)

**Software**, also called a **program**, consists of a series of instructions that tells the computer what tasks to perform and how to perform them.

**System software** programs that coordinates all activities among computer hardware devices and allow the user to perform maintenance type tasks usually related to managing a computer, its devices or its programs Example: operating system, disk defragmenter, etc

**Application software** programs that perform a specific task and are designed to make users more productive

An **operating system** (OS) is a set of programs containing instructions that work together to coordinate all the activities among computer hardware resources.

**Analog signals:** continuous wave forms

**Digital signals:** represent information as a sequence of 0s and 1s (binary code or binary number

**Machine language:** language of a computer

**Binary digit (bit) bit):** the digit 0 or 1

**Byte:** a sequence of eight bits

ASCII (American Standard Code for Information Interchange) the most widely used coding scheme to represent data - 128 characters (0 to 127), extended version: 256 characters

EBCDIC Extended Binary Coded Decimal Interchange Code) Used by IBM 256 characters (1 byte)

Unicode: 65536 characters (2 bytes) - capable of representing all world’s languages

Computer program - series of instructions that directs computer to perform tasks

Programming language - used to communicate instructions to a computer

Low-level language: Machine dependent runs only on one type of computer - Machine and assembly languages are low level - Programmers need to know the instruction set of the microprocessor - First generation language (1GL) - Second generation language (2GL)

High-level language: Often machine independent can run on many different types of computers and operating systems -Programs (source code) are develop faster, have fewer errors, and are easier to read - Third generation language (3GL) - Fourth generation language (4GL) - Fifth generation language (5GL)

Machine language: Only language computer directly recognizes - Uses a series of binary digits (1s and 0s) with a combination of numbers and letters that represent binary digits (binary code) - Early computers were programmed in machine language

Assembly language: Instructions made up of symbolic instruction codes, meaningful abbreviations and codes - Source program contains code to be converted to machine language by an **assembler**

Programmer writes instructions (**source program**) that tell computer what to accomplish and how to do it

**Compiler** translates an entire program before executing it

**Interpreter** converts and executes one code statement at a time

Spaghetti code: Early procedural languages had line numbers and “goto” statements

Structured design: dividing a problem into smaller subproblems (approach is also called top down (or bottom up) design, stepwise refinement, or modular programming)

**Object** is a single unit component that contains data and operations on data

**Nonprocedural Programming Language:** The programmer writes English like instructions or interacts with a visual environment to retrieve data from files or a database

**Program Development Tools**: User friendly environment designed to assist both programmers and users in creating programs

**Visual programming environment** (VPE) allows developers to drag and drop objects to build programs

**Visual Programming Languages** provide visual or graphical interface for creating source code

**Event driven** checks for and responds to set of events (actions to which programs can respond to)

Programming is a process of problem solving

Problem-solving technique: –Analyze the problem–Outline the problem requirements–Design steps (algorithm) to solve the problem

Algorithm - a step by step problem solving process that reaches a solution in a finite amount of time

1.Analyze the problem–Outline the problem–Outline the problem requirements–Design steps (algorithm) to solve the problem 2.Implement the algorithm–Implement the algorithm in code–Verify that the algorithm works 3.Maintain the program–Use and modify the program if the problem domain changes

***Chapter 1a 1b***

The process of walking through a program’s logic on paper before you write the program is called **desk checking**

A **structure** is a basic unit of programming logic–Sequence–Selection–Repetition/Loop

**Decision symbol:** a diamond shape in a flowchart that represents a decision/question

Action or actions that occur within the loop are known as the **loop body**

Programmers refer to looping as **repetition** or **iteration**

**Infinite loop:** repeating flow of logic with no end

Attaching structures end to end is called **stacking structures**

Placing a structure within another structure is called **nesting structures**

***Chapter 2***

1. **Editor:** Use an editor to create a source program in C++

2. **Preprocessor:** Include the code from the preprocessor directives

3. **Compiler:** Use the compiler to: ▪ Check that the program obeys the rules ▪ Translate into machine language (object program)

4. **Linker:** Combines object program with other programs and libraries provided by the software development kit (SDK) to create **executable code**

5. **Loader:** Loads executable program into main memory

6. **Execute** the program

**Function:** collection of statements; when executed, accomplishes something – May be predefined or standard

**Syntax:** rules that specify which statements (instructions) are legal

**Semantic rule:** meaning of the instruction

**Token:** the smallest individual unit of a program written in any language

**Data type:** set of values together with a set of operations

Three categories of simple data ▪ **Integral:** integers (numbers without a decimal) ▪ **Floating-point:** decimal numbers ▪ **Enumeration type:** user-defined data type

A **string** is a sequence of zero or more characters

**Named constant:** memory location whose content cannot change during execution

**Variable:** memory location whose content may change during execution

**=** is called the **assignment operator**

**Increment operator (++):** increment variable by 1

**Decrement operator (--):** decrement variable by 1

**Compound assignments:** combination of assignment and arithmetic operator

**cin** is used with >> (the **stream extraction operator**) to gather input/read data

**cout** and << (**the stream insertion operator**) are used to output the value of an expression at the current cursor position or format the output

A **manipulator** is used to format the output ▪ **endl** causes the insertion point to move to beginning of next line

**Escape sequences** are used to display characters that have a special meaning in C++ or in an output statement

**Preprocessor directives** are commands supplied to the preprocessor to include the header files for these libraries

▪ A C++ program is a collection of functions, one of which is the function main

▪ The first line of the function main is called the heading of the function: int main()

▪ The statements enclosed between the curly braces ({and}) form the body of the function – Contains two types of statements: ▪ **Declaration statements:** declare things, such as variables ▪ **Executable statements:** perform calculations, manipulate data, create output, accept input, etc.

**Prompt lines:** executable statements that inform the user what to do

**;** is called **statement terminator**

***Chapter 3***

**Stream:** sequence (stream) of bytes from source to destination ▪ Bytes are usually characters, unless program requires other types of information

**Input stream:** sequence of characters from an input device to the computer

**Output stream:** sequence of characters from the computer to an output device

Use **iostream** header file to extract (receive) data from keyboard and send output to the screen

**Function (subprogram):** set of instructions ▪ When activated, it accomplishes a task

**main** executes when a program is run

Other functions execute only when called

C++ includes a wealth of functions

**Predefined functions** are organized as a collection of libraries called header files that may contain several functions

The **get** function ▪ Inputs next character (including whitespace) ▪ Stores in memory location indicated by its argument

**Dot** separates the input stream variable name from the member or function name

In C++, dot is the **member access operator**

**ignore** discards a portion of the input

**Putback** places character back to the input stream

**peek** returns next character from the input stream, without removing the character from that stream

The **clear** function restores input stream to a working state (not in Visual C++)

The function **getline** - Reads until end of the current line - Read an entire line (until ‘\n’) from cin istream into the variable VarStr - Can be used for any istream for reading strings that contain non-newline whitespaces (space,tab)

**Setprecision Manipulator** - Outputs decimal numbers with up to n decimal places - Must include the header file iomanip

**fixed** outputs floating-point numbers in a fixed decimal format

**scientific** outputs floating-point numbers in scientific format

**showpoint** forces output to show the decimal point and trailing zeros

**setw Manipulator:** Outputs the value of an expression in specific columns – Output of the expression is right-justified – Unused columns to the left are filled with spaces

**setfill Manipulator:** Output stream variables can use **setfill** to fill unused columns with a character

**left and right Manipulators:** left-justifies the output / right-justifies the output

**Parameterized manipulator:** with parameters – require iomanip and iostream hearders – setprecision, stew, and setfill

**Nonparameterized manipulator:** without parameters – require iostream header – endl, fixed, scientific, showpoint, left, and right

**File Input/Output:** File is an area in secondary storage to hold information

***Chapter 4***

logical (Boolean) expressions: an expression that has a value of either true or false

conditional expression: an expression that uses a conditional operator

compound statement: consists of a sequence of statements enclosed in curly braces, { and }

logical (Boolean) values: true and false

conditional operator: a ternary operator written as “?:”; the three arguments explain what the condition is, what the result will be if the condition is true, and what the result will be if the condition is false

Logical (Boolean) operators: operators that enable you to combine logical expressions

action statement: the statement following the expression in an if statement

pseudocode: an informal mixture of C++ and ordinary language used to design an outline of a logical solution to a problem

decision maker: the expression in an if statement which determines whether to execute the statement that follows it

nested: when one control statement is located within another

***Chapter 5***

infinite loop: a loop that continues to execute endlessly

loop control variable (LCV): a variable that controls the end of the loop

counter-controlled while loop: a while loop that is used when you know how many items of data are to be read; the loop will continue until the condition designated by the counter is met or evaluates to false

flag-controlled while loop uses a Boolean variable to control the execution of the loop

flag variable: a Boolean variable used to control the execution of a while loop

end-of-file (EOF)-controlled while loop: a while loop that stops when it reaches the end of the input file

Fibonacci number: a number in the Fibonacci sequence

Fibonacci sequence: a n = a n-1 + a n-2

for loop control: a loop control variable in a for loop; also called an indexed variable

nesting: a process that involves putting one control structure inside another

divisor: suppose that m and n are integers and m is nonzero. Then m is called a divisor of n if n = mt for some integer t; that is, when m divides n, the remainder is 0