Chapter 1: Introduction to Computers and Programming

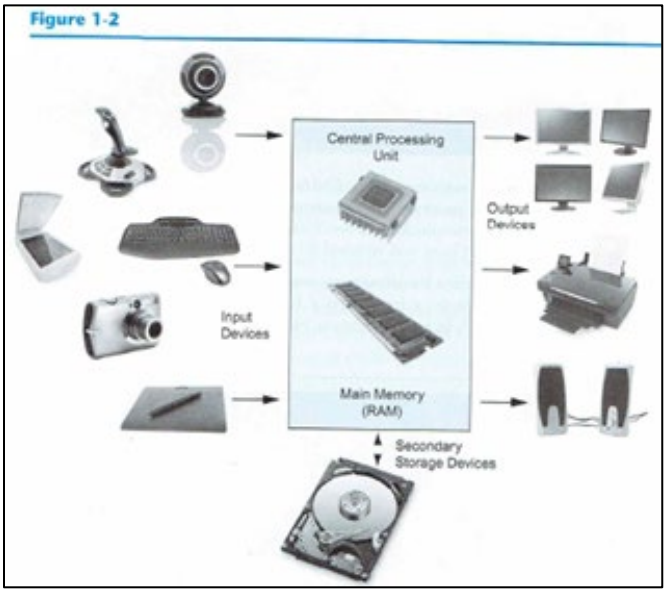
Section 1.2: Hardware and software

Hardware

A typical computer System consists of:

- **The CPU**: Process data
- **Main memory**:
- **Secondary Storage**: This can hold memory for long period
- **Input devices:** The device that collects the information and send it to the computer is called an input device.
- **Output devices**: The information is sent to an output device.

Organization of a computer system.

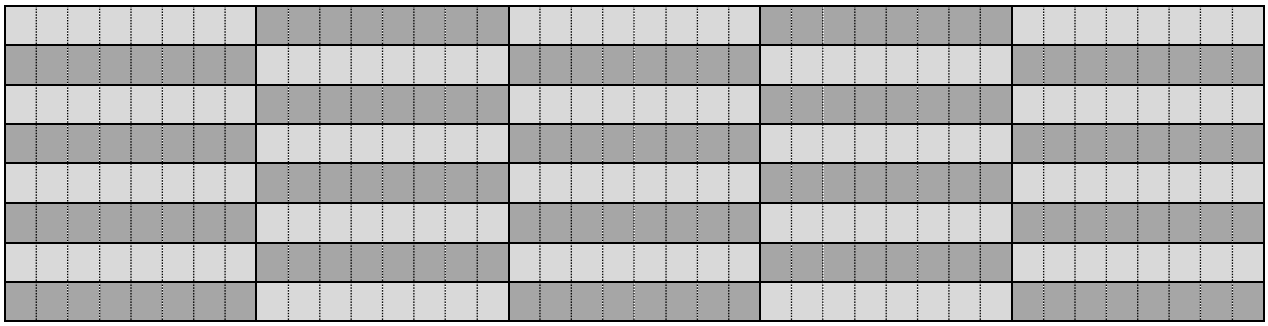


Main Memory

- The _____ and the _____ are stored here in main memory.
- Main memory is known as **RAM (** _____ **)** RAM is a volatile type of memory; only for temporary memory while program is run.

- Computer memory is divided into tiny storage locations as _____; Each byte is assigned an address.
- One byte is enough memory to store only a letter of the alphabet or a small number.
- Each byte is divided into _____ smaller storage locations called _____; Bit stands for binary digit.

A visual Representation of memory



Software

Two general categories of software:

- **Application software**: The programs that makes the computer useful.
  e.g., Word, Excel, photoshop
- **System software**: The programs that control and manage the basic operations of a computer.
  e.g., Operating systems, Utility programs, and Software development tools

Programming language and Machine language

A **Programming Language** is a special language used to write computer programs. Programming language uses words instead of numbers; the program written in a programming language is called the source code.

A **Machine Language** program consists of stream of binary numbers (i.e., 0 and 1s); A CPU can only process instructions that are written in machine language. A program in machine language is called binary code.

Compilers are used to convert a source code into the binary code.

Programs and programming language

1. Write the instructions (in English) to calculate the gross pay (refer pg 8).

   Step 1. Display the message: "How many hours did you work?"

   Step 2. Wait for the user to enter the number of hours and store it in memory.

   Step 3. Display the message: "How much do you get paid for an hour?"

   Step 4. Wait for the user to enter the hourly pay rate and store it in memory.

   Step 5. Calculate the gross pay and store it in memory.

   Step 6. Display a message that tells the gross pay.

Collectively, these instructions are called an algorithm or program.

2. Write a program (in English) to calculate the difference of two numbers; both numbers should be taken as keyboard inputs.

Step 1. .....

Step 2. .....

Step 3. .....

Step 4. .....

Step 5. .....

Step 6. .....

3. Write a program (in English) to display the message: “Hello World”.

Step 1. .....

Step 2. .....

Programming languages

There are two categories of programming languages:

- low-level language: closed to level of machine language.
- High-level language: closed to human’s natural language. E.g., Java , C++, Visual Basic, C#, C, Python.

1. Write a program (in C++) to display the message: “Hello World on the screen: helloWorld.cpp

```
//**************************************************************************
//
//               File:                    helloWorld.cpp
//
//               Student:                 Lasanthi Gamage
//
//               Assignment:              Program  #1
//
//               Course Name:             Programming   I
//
//               Course Number:           COSC 1550 - 01
//
//               Due:                     Jan 24, 2019
//
//
//               This program outputs the message "Hello World" on the screen
//
//**************************************************************************

#include <iostream>
using namespace std;

int main()
{
        cout << "Hello World" << endl;

        return 0;

}
```

2. Write a program that displays your name (in C++): name.cpp

```
//**************************************************************************
//
//              File:                   name.cpp
//
//              Student:                Lasanthi Gamage
//
//              Assignment:             Program  #2
//
//              Course Name:            Programming   I
//
//              Course Number:          COSC 1550 - 01
//
//              Due:                    Jan 24, 2019
//
//
//              This program outputs the name of the Author on the screen
//
//**************************************************************************


#include <iostream>
using namespace std;

int main()
{

.....................................................................................................


.....................................................................................................


       return 0;

}
```

3. Write a program that calculates the gross pay and display on the screen: grossPay.cpp.

Figure 1- 1: Program 1-1 (Rewritten according to department standards)

```
//**************************************************************************
//              File:                   grossPay.cpp
//
//              Student:                Lasanthi Gamage
//
//              Assignment:             Program  #
//
//              Course Name:            Programming   I
//
//              Course Number:          COSC 1550 - 01
//
//              Due:                    Jan 24, 2018
//
//
//              This program calculates the gross pay and display on the
//              screen
//
//**************************************************************************

#include <iostream>
using namespace std;

int main()
{
    double  hours,
            rate,
            pay;
```

Step 1. Display the message: "How many hours did you work?"

Step 2. Wait for the user to enter the number of hours and store it in memory.

Step 3. Display the message: "How much do you get paid for an hour?"

Step 4. Wait for the user to enter the hourly pay rate and store it in memory.

Step 5. Calculate the gross pay and store it in memory.

Step 6. Display a message that tells the gross pay.

```
    cout << "How many hours did you work? ";
    cin >> hours;


    cout << "how much do you get paid per hour? ";
    cin >> rate;

    pay = hours * rate;                                  //Calculate the pay.

    cout << "You have earned $" << pay << endl;          //Display the pay

    return 0;
}
```

Common elements of a programming language

Element	Examples
Key words	.....
Program-defined identifiers	.....
Operators	.....
Punctuation	.....
Syntax	.....

Lines vs Statements

Line: a single line appears in the program body.
Statement: a complete instruction that causes the computer to perform something.

Example 1: In program 1-1, each statement takes only one line.
Example 2:

```
cout << "How much do you get paid per hour?
";
```

Lines: .....

Statements: .....

Example 3:

```
cout << "How much do you get"
        << " paid per hour? ";
```

Lines: .....

Statements: .....

Example 4:

```
cout << "How much do you get " ;
cout<< "paid per hour? ";
```

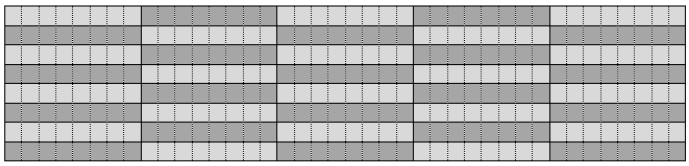
Lines: .....

Statements: .....

Review Programming Style Guide (pg 5 of 9): Line length and Indentation of Continued Lines

Variables

A variable is a named (symbolic) _____. E.g., _____.
The information stored in a variable is stored in RAM.



Variable definitions

Two general types of data: _____ and _____. Numbers are used to perform mathematical operations. Characters are used to print data on the screen or on a file or on paper.

Numeric data can be categorized even further. For example, _____ numbers and _____ (floating point) numbers.

When creating a variable in C++, you must know the type of the data the program will be storing in.
ex. double hours,
rate,
pay; // variable definition
The variable definition comes **BEFORE** any other statements using those variables.

Review Programming Style Guide (pg 6 of 9): Variable Declaration Style

Exercise:

Define a variable to store a whole number (the type is *int*) named *numStudents*.

```
int   numStudents;
```

Define a variable to store a whole number (the type is *int*) named *test1Score*.

.....

Define a variable to store a real number (the type is *float*) named *assgment1*.

.....

Define a variable to store a real number (the type is *float*) named *avgScore*.

.....

Define a variable to store a single character (the type is *char*) named *letterGrade*.

.....

Define two variables to store real numbers (the type is float) names *num1* and *num2*.

.....

.....

1.5: input, processing and output.

Three primary activities of a program: input gathering, performing some process on the information gathered, and producing output.

Input

Input is information a program collects from the outside world. The ways to send this information to the program: as keyboard input, using mouse, reading a file, etc. In a program, an input statement should always follows an output statement which prompts for inputs.

Exercise:

Write a statement to read a number from the keyboard and store it in a variable (you may assume the variable definition: ***int numStudents;***)

cin >> numStudents;

Write a statement to read a number from the keyboard and store it in a variable (you may assume the variable definition: ***int test1Score;***)

.....

Write a statement to read a character from the keyboard and store it in a variable (you may assume the variable definition: ***char initial;***)

.....

Write a statement to read a number and a character from the keyboard and store it in a variable (you may assume the variable definition: ***int numStudents;, and char initials;***)

.....

.....

.....

Processing

Once information is gathered, the program usually processes it. (note: there is only one and only one variable on the left-hand side of the equal sign.)

Exercise:

Write a statement to find the half of a number (stored in *original*) and store the result in *discount*.

discount = *original / 2;*

Write a statement to find the twice of a number (stored in *age*) and store the result in *ageTwice*.

.....

Write a statement to find the average of three test scores (stored in *test1Score*, *test2Score*, and *test3Score*). Store the result in *avgScore*.

.....

Output

Output is information that a program sends to the outside world. It can be something you display on a screen, file output, or a report send to the printer.

Exercise:

Write a statement to display a message on the screen that tells the discount amount.

cout << "Your discount is: " << discount*;*

Write a statement to display a message on the screen that tells the twice of the age.

.....

Write a statement to display a message on the screen that tells average of three test scores.

.....

Combining processing and output (**good vs bad programming.)**

The following two statements process data and output the result on the screen.

discount = original/2;

cout << "Your discount is: " << discount;

In terms of the output, the above code segment is equivalent to the following, but bad programming (not recommended to use in your programs):

cout << "Your discount is: " << original/2;

HW1:

(Submit through WorldClassroom)

Write a C++ program **SEGMENT** that takes two keyboard inputs, which are two sides of a rectangle, find the area of the rectangle, and display the result on the screen. Your code should contain separate statements for variable definition, input, processing, and output.

1. .....
2. .....
3. .....
4. .....
5. .....
6. .....
7. .....
8. .....
9. .....
10. .....
11. .....
12. .....
13. .....
14. .....
15. .....
16. .....
17. .....
18. .....

HW 2:

(Submit through WorldClassroom)

Write a C++ program segment that takes two user inputs (for two numbers) and add, subtract, multiply, and divide the two numbers, and output the results. Your code should contain separate statements for variable definition, input, processing, and output.

1. .....
2. .....
3. .....
4. .....
5. .....
6. .....
7. .....
8. .....
9. .....
10. .....
11. .....
12. .....
13. .....
14. .....
15. .....
16. .....
17. .....
18. .....
19. .....
20. .....
21. .....
22. .....
23. .....

Lengths of lines in a program

Figure 1-1: Program 1-1 (Rewritten according to department standards)

```
//*************************************************************************

//*************************************************************************
//              File:                   grossPay.cpp
//
//              Student:                Lasanthi Gamage
//
//              Assignment:             Program  #
//
//              Course Name:            Programming   I
//
//              Course Number:          COSC 1550 - 01
//
//              Due:                    Jan 24, 2018
//
//
//              This program calculates the gross pay and display on the
//              screen
//
//*************************************************************************

#include <iostream>
using namespace std;

int main()
{
    double  hours,
            rate,
            pay;


    cout << "How many hours did you work? ";
    cin >> hours;                                   //Get number of hours
                                                    //worked.

    cout << "how much do you get paid per hour?";
    cin >> rate;

    pay = hours * rate;                             //Calculate the pay.

    cout << "You have earned $" << pay << endl;     //Display the pay
    return 0;
}
```

1st column - left most code starts here

4th column - First level of indentation Starts here

85th column - All comments start here.

103rd column; No character should pass this column