Republic of the Philippines
POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
COLLEGE OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

CMPE 30022

Programming Logic and Design

LAB EXERCISE

4

INPUT AND VARIABLES IN PYTHON

Submitted by:

Pascual, Aaron Raphael L.

95%

Year and Section:

BSCPE 1-1

Submitted to:

Sir Julius Cansino

Date Submitted:

August 16, 2019

Python Activity 4: Input and Variables in Python

"How do you input and store data in a Python program?"

Learning Objectives

Students will be able to: ***Content:***

- Explain how to input data in Python
- Explain the meaning and purpose of a variable
- Determine if a variable name is valid
- Explain concatenation and the use of "+"

Process:

- Create **input** statements in Python
- Create *Python* code that prompts the user for data and stores it in a variable
- Create valid and good variable names

Prior Knowledge

- Understanding of flowchart symbols

Further Reading

- https://www.learnpython.org/en/Variables_and_Types
- https://www.afterhoursprogramming.com/tutorial/python/variables-py/
- https://en.wikibooks.org/wiki/Non-Programmer%27s_Tutorial_for_Python_3/Who_Goes_There%3F

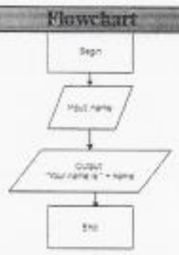
Information:

This activity asks you to execute lines of Python Code. Start the Thonny IDE. In the Shell window you can type individual command lines, or in the editor window you can type a whole program and run it.

Model 1: Input in *Python*

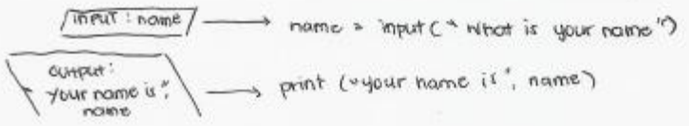
Input, one of the four main operations of a computer, is performed using an **input** statement in **Python**. The **prompt** that appears on the screen to tell the user what to enter is included as a *string literal* in the parentheses of the **input** statement. Input( ) and print( ) are known as ***functions*** in python.

Flowchart	Python Program

	`# Programmer: Monty Python`
	`# Date: Sometime in the past`
	`# Description: A program that introduces variables`
	
	`name = input("What is your name? ")`
	`print ("Your name is", name)`

Critical Thinking Questions:

1. Enter and execute the Python program using the editor window in Thonny. To run the program when you are done typing you can either press F5, or click the green arrow . What is printed in the shell window when the Python program is executed?



2. Draw a line between each flowchart symbol and its corresponding Python code.

3. Examine the first line of Python program: `name = input("What is your name? ")`

   a. What happens when this line of code is executed? this line will let you input/type your name with a question "what is your name?"

   b. What appears in the console window when this line of code is executed?

   the prompt will appear, "What is your name?"

FYI: The words that appear on the screen to tell the user what to enter is known as a ***prompt***.

   c. Where is the data stored when the user enters something and presses the Enter button?

   the data will be stored in the variable

4. Notice that the word **name** is in both lines of code. Is the string literal **"name"** printed when the second line of Python code is executed? What is printed?

no, the data that will appear is what the user input

Model 2: Variables in *Python*

The word **name** in the Python code is a ***variable*** – *name given to a memory location used to store data.*

```
name = input("What is your name? ")
print("Your name is", name)
```

5. What happens when you execute each of the following lines of Python code?

   a. `name? = input("What is your name?")`

   invalid syntax, because there is a question mark

   b. `your name = input("What is your name?")`

   invalid syntax, because there is a space

   c. `1st_name = input("What is your name?")`

   invalid syntax, because the ~~first~~ variable starts w/ a number

   d. `from = input("Where were you born?")`

   invalid syntax, because it is a keyword

6. Examine the errors that occurred when executing the lines of code in question 5. Then examine the following lines of valid code.

```
name2 = input("What is your name?")
your_name = input("What is your name?")
yourName = input("What is your name?")
```

What are the rules for creating a valid ***variable*** name?

the rules are we must not have space, starts with a number, a keyword, and a question mark for it to function well.

7. Are the following variable names valid? Are they good names? Why or why not?

Variable name	Comments about variable name
price	yes, because it is simple and summarizes the thing that is really needed.
costoffirstitem	yes, for we can eatily see and calculate the cost of all items.
1total	no, because it starts with a number.
firstName	yes, because it doesn't violate any rule and this will avoid the misunderstanding in terms of numbers.
Qty2!	no, because it has an exclamation point.
Quiz Score1	no, because it has a space.

Information: Printing Multiple Items and Concatenation
You can concatenate two strings literals, two string variables or a string literal and a string variable using the "+" symbol. The strings can be string literals or a variable containing string literals. To print multiple values without being concerned whether they are strings, you can use a "," comma.

8. Predict the output and execute the following lines of code.

```
File Edit Format Run Options Windows Help
name = input("What is your name? ")
print("Your name is", Name)
                                    Ln: 3 Col: 0
```

a. Is the output what you would expect? Why or why not?
no, because in the keyword area the word Name starts with a capital "N" since the variable starts with a small n, it will not function properly.
b. How can you alter the code so that it functions properly?
by simply changing the variable "name" by a capital "N"

9. Use the following set of Python statements to answer the questions below.

```
File Edit Format Run Options Windows Help
print("Your name is", "Pat.")
print('Your name is', "Pat.")
print("Your name is" + "Pat.")
print("Your age is", 20)
print("Your age is" + 20)
```

a. State the output for each of line of code.

1. your name is Pat.
2. your name is Pat.
3. your name is Pat.
4. your age is 20
5. Error / Traceback (most recent call back)

b. What is the difference between the first two print statements? Does the difference affect the output?

it differs in terms of the quotation but does not make any difference neither affect the output.

c. Notice that some statements include a comma (,) between the two literals being printed and some statements use a "+". Do they produce the same output?
No, because using comma will seperate the words while using + sign will make the 2 words into 1 word.

d. Explain the purpose of the comma.

the purpose of comma is to seperate or make a space after the words that are inside of the quotation marks.

e. Why does the last print statement crash the program? What would you do to correct it?

in order to correct the last statement, we must put a quotation marks in the integer "20".

10. Execute the following program:

```
name = input("Enter your name: ")
ID = input("Enter your student ID number: ")
course = input("Enter your course number: ")

print(name + "'s ID is " + ID +"\nand is enrolled in " + course)
```

a) State what is displayed on the screen when you executed the program.
Enter your name:
Enter your student ID number:
Enter your course number:
"name's ID is "<the ID no.>"
"and is enrolled in "<course number>"

b) The \n is a "string literal escape sequence" in python. What does this do?

its purpose is to translate that to the proper newline character for your platform.

c) Change the \n to \t and describe what happens to the program output.

its used for tab character and makes a horizontal Tab space between two strings or characters.

Application Questions: Use the Thonny IDE to check your work

1. State a good variable name for an employee's ID number.

EmployeeIDnumber

2. Provide an example of a string literal.

\" monty python"

3. Provide an example of a numeric literal.

(- or +), 0 - 9

4. Write a line of Python code that prompts the user for the name of their favorite ice cream and stores it in a valid variable name. icecream = input(" Enter your favorite ice cream ")

More Programming Activity

ACTIVITY 4.1 Write a Python program that prompts the user for their first name and last name. Assuming that the user's first name is "Monty" and their last name is "Python" the output should be:

```
Enter your first name:Monty
Enter your last name:Python
Monty's last name is Python.
```

fname = input ("Enter first name:")
lname = input ("Enter last name:")
print (fname + "'s last name is " + lname + ".")

Be sure your output has the same punctuation as the above output.

ACTIVITY 4. 2 Crazy Sentence Program. Create a program that prompts the user for the name of an animal, a color, the name of a vehicle, and the name of a city. Then print a sentence that contains the user input in the following order. Include the additional words in the sample output as part of your output. Example: Assume the user enters the words: tiger, green, motorcycle, and Wildwood. The output would be:

```
The green tiger drove the motorcycle to Wildwood.
```

ACTIVITY 4.3 Design a program that displays the following information:

- Your name
- Your address, with city, state, and ZIP
- Your telephone number
- Your college major

ACTIVITY 4.1

```
5 fname = input ("Enter your first name:")
6 lname = input ("Enter your last name:")
7 print (fname + "'s last name is " + lname + ".")
```

```
>>> %Run 'lab exercise 4.py'

  Enter your first name:Monty
  Enter your last name:Python
  Monty's last name is Python.
```

ACTIVITY 4.2

```
5 animal = input ("Enter name of an animal:")
6 color = input ("Enter name of a color:")
7 vehicle = input ("Enter name of a vehicle:")
8 city = input ("Enter name of a city:")
9 print ("The",color,animal,"drove the",vehicle,"to",city + ".")
```

```
>>> %Run 'lab exercise 4.py'

  Enter name of an animal:tiger
  Enter name of a color:green
  Enter name of a vehicle:motorcycle
  Enter name of a city:Wildwood
  The green tiger drove the motorcycle to Wildwood.
```

ACTIVITY 4.3

```
5  name = input ("Enter your name:")
6  city = input ("Enter your city:")
7  address = input ("Enter your address:")
8  state = input ("Enter your state:")
9  code = input ("Enter your zipcode:")
10 number = input ("Enter your telephone number:")
11 cmajor = input ("Enter your college major:")
12 print ("Your name is:",name)
13 print ("Your address is:",address+",",city+",",state+",",code)
14 print ("Your telephone number is:",number)
15 print ("Your college major is:",cmajor)
```

```
>>> %Run 'lab exercise 4.py'

  Enter your name:Kane Edward Y. Malapo
  Enter your city:Makati City
  Enter your address:105-C 6th Avenue
  Enter your state:Philippines
  Enter your zipcode:1216
  Enter your telephone number:800-7922
  Enter your college major:Computer Engineering
  Your name is: Kane Edward Y. Malapo
  Your address is: 105-C 6th Avenue, Makati City, Philippines, 1216
  Your telephone number is: 800-7922
  Your college major is: Computer Engineering
```

Republic of the Philippines
POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
COLLEGE OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

CMPE 30022

Programming Logic and Design

LAB EXERCISE 1

COMPUTER HARDWARE

Submitted by:

Robin I. Pasaldedos

Submitted to:

Engr. Julius Cansino

Date Submitted:

June 25, 2019

LAB 1: Computer Hardware

Duration: 3 Hours

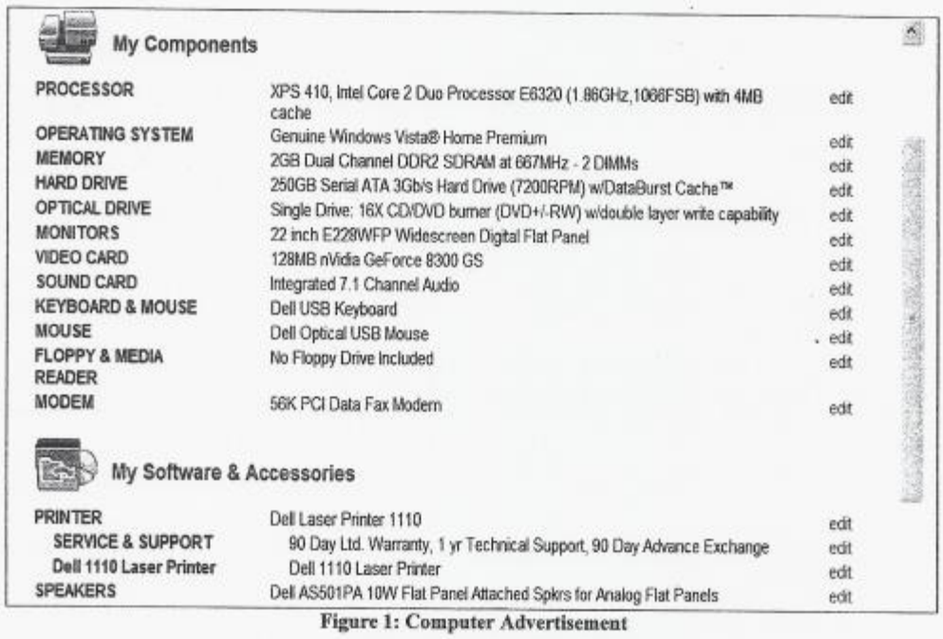
Objectives:

- Understand computer hardware.
- Understand basic computer components.
- Learn the function of storage.
- Learn about history of computers.

Task 1: Computer Components

As previously mentioned, the computers that are used nowadays are based upon the stored-program computer concept proposed by John Von Neumann. Also, in Module 1, we pointed out that a computer is a device capable of ***storing***, ***retrieving***, and ***processing information*** or ***data***. These suggest that a computer must consist of four basic components. These components are

- Central processing unit (CPU)
- Memory unit
- Input unit
- Output unit



My Components

PROCESSOR	XPS 410, Intel Core 2 Duo Processor E6320 (1.86GHz,1066FSB) with 4MB cache	edit
OPERATING SYSTEM	Genuine Windows Vista® Home Premium	edit
MEMORY	2GB Dual Channel DDR2 SDRAM at 667MHz - 2 DIMMs	edit
HARD DRIVE	250GB Serial ATA 3Gb/s Hard Drive (7200RPM) w/DataBurst Cache™	edit
OPTICAL DRIVE	Single Drive: 16X CD/DVD burner (DVD+/-RW) w/double layer write capability	edit
MONITORS	22 inch E228WFP Widescreen Digital Flat Panel	edit
VIDEO CARD	128MB nVidia GeForce 8300 GS	edit
SOUND CARD	Integrated 7.1 Channel Audio	edit
KEYBOARD & MOUSE	Dell USB Keyboard	edit
MOUSE	Dell Optical USB Mouse	edit
FLOPPY & MEDIA READER	No Floppy Drive Included	edit
MODEM	56K PCI Data Fax Modem	edit

My Software & Accessories

PRINTER	Dell Laser Printer 1110	edit
SERVICE & SUPPORT	90 Day Ltd. Warranty, 1 yr Technical Support, 90 Day Advance Exchange	edit
Dell 1110 Laser Printer	Dell 1110 Laser Printer	edit
SPEAKERS	Dell AS501PA 10W Flat Panel Attached Spkrs for Analog Flat Panels	edit

Figure 1: Computer Advertisement

Activity 1.1: Complete the following table by listing the actual components from the ad in Figure 1 to match the categories listed.

Basic Component	Component from Figure 1
Central Processing Unit	Operating systems
Memory Unit	memory, Hard drive, optical drive, video card, Floppy & media,
Input Unit	mouse, keyboard & mouse,
Output Unit	monitors, printer, speaker, modem, sound card video card

Processor

Activity 1.2: A central processing unit or processor consists of ***Arithmetic Logic unit*** (ALU) and ***Control unit*** (CU) unit. What is the function of ALU?

The function of ALU is to perform arithmetic and logic operations.

Activity 1.3: What is the function of control unit?

Control Unit transfers data between CPU or microprocessor and ALU.

Activity 1.4: What does "***Core 2 Duo Processor***" in the description of processor in Figure 1 mean?

Two processor cores work inside a Core 2 Duo in parallel

Processor Speed

What does "***1.86 GHz***" in the description of processor in Figure 1 mean?

GHz stands for **giga-Hertz**, which means ***billion cycles/ticks per second***. This is a unit used to measure CPU/processor or clock speed. Clock speed is the speed at which the processor executes instructions. The CPU requires a fixed number of clock cycles/ticks to execute each instruction. The faster the clock, the more instructions the CPU can execute per second.

So, ***1.86 GHz*** refers to the processor that can process instructions at the speed of 1.86 billion cycles per second.

Clock speed can be used as a rough comparison of the speed of two processors as long as they are the same type and the same brand/maker. Comparing different processors is much more difficult. The clock still makes each processor do something each cycle, but what that something is can be very different. For example, on old computers, it would take several cycles for the processor to complete one operation. On a Pentium4, however, it usually completes two operations per cycle.

Figure 2: Description of Two Intel Precessors

Activity 1.5: In the above Figure 2, what does "*2.4 GHz*" in the description of processor *Intel core 2 Duo Processor E6600* mean?

2.4GHz can process instructions at the speed of 2.4 billion cycles per second.

Activity 1.6: In the above Figure 2, which processor is faster, E6320 or E6600? Why?

E6600, because GHz of it is higher than E6320.

Task 2: Memory

What is computer memory or memory? From:

1. The American Heritage® Dictionary: ***Memory*** *is*
   a. *A unit of a computer that preserves data for retrieval.*
   b. *Capacity for storing information: two gigabytes of memory.*
2. The Encyclopedia Britannica: ***Computer Memory*** *is device that is used to store data or programs (sequences of instructions) on a temporary or permanent basis for use in an electronic digital computer.*

From the Webopedia Computer Dictionary, "*Every computer comes with a certain amount of physical memory, usually referred to as* ***main memory*** *or* ***RAM****. You can think of main memory as an array of boxes, each of which can hold a single byte of information. A computer that has 1 megabyte of memory, therefore, can hold about 1 million bytes (or characters) of information.*"

Activity 2.1: What does RAM stand for?

Random access Memory

Activity 2.2: There are two basic types of RAM: ***Dynamic RAM*** (DRAM) and ***Static RAM*** (SRAM). What is a ***Dynamic RAM***?

Dynamic RAM stores bits in cells consisting of a capacitor and a transistor.

Activity 2.3: What is a ***Static RAM***?

Static RAM retains data bits in its memory.

Activity 2.4: Which is more expensive ***Dynamic RAM*** or ***Static RAM***?

Static RAM because its better than Dynamic RAM based on performance.

Activity 2.5: When someone says a computer has "one gigabyte of RAM", do they mean the computer has one gigabyte of ***Dynamic RAM*** or ***Static RAM***?

Yes, by checking the specification of this memory.

My Components

PROCESSOR	XPS 410, Intel Core 2 Duo Processor E6320 (1.86GHz, 1066FSB) with 4MB cache
OPERATING SYSTEM	Genuine Windows Vista® Home Premium
MEMORY	2GB Dual Channel DDR2 SDRAM at 667MHz - 2 DIMMs

Figure 3: Description of Memory

Activity 2.6: Is the memory listed in **My Components** in Figure 3 of type ***Dynamic RAM*** or ***Static RAM***?

Dynamic RAM

Activity 2.7: What is "***SDRAM***" in the description of **Memory** in Figure 3?

SDRAM stands for Synchronous Dynamic Random Access Memory that synchronized with the clock speed that the microprocessor is optimized for.

Activity 2.8: What does "***DDR2***" in the description of **Memory** in Figure 3 stand for?

Double Data Rate 2

Activity 2.9: What is "***cache***" in the description of **Processor** in Figure 3?

Cache reduce the average cost (time or energy) to access data from the main memory.

Activity 3.10: Is **cache** a memory of type ***Dynamic RAM*** or ***Static RAM***?

Static RAM

Activity 2.11: How many types are of **cache**? What are they?

There are 3 types of memory, L1 cache or primary cache, L2 cache or Secondary cache and L3 cache or Main memory.

Task 3: Input and Output and Secondary Storage

Input and Output

Input and output are the basic computer components that communicate with CPU. An input device sends data/information to CPU. An output device takes the processed data/information from CPU and makes it available for the user.

Activity 3.1: List all input devices that you know?

mouse, keyboard, microphone, Web cam, scanner

Activity 3.2: List all output devices that you know?

Monitor, speaker, printer, Projector, plotter,

Activity 3.4: What is a USB/flash drive?

USB flash drive is a storage device that includes flash memory with an integrated USB interface.

Activity 3.5: Is USB/flash drive an input or output device?

It can use as input device and also as output device

Activity 3.6: What does USB stand for?

Universal Serial Bus

Activity 3.7: What is secondary storage?

Secondary storage is a storage device and media that are not constantly accessible by a computer system

Activity 3.8: Is USB/flash drive a secondary storage?

Yes, because its not accessible by a computer system.

Activity 3.9 List all secondary storages that you know?

Floppy diskette, Hard drive, CD-ROM Disc, USB drives, Memory cards,

Activity 3.10 What do CD and DVD stand for?

Compact Disc & Digital Versatile Disc

Activity 3.11 Are CD and DVD secondary storage?

yes, they are Optical storage devices.

Activity 3.12 What is a capacity of a CD?

The capacity of a CD is up to 700MB

Activity 3.15 What is a capacity of a DVD?

The capacity of a DVD is up to 4.3 GB

Activity 3.16 What is a CD-R?

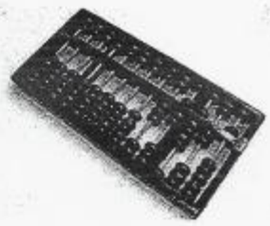
A blank CD which can be recorded on once only.

Activity 3.17 What is a CD-RW?

A blank CD that can be recorded, erased, and rerecorded many times.

Task 4: History of Computers

The first computing device is ***Abacus*** which can be traced back to 5,000 years ago in Asia and still is being used today. **Abacus** is considered to be a mechanical computing device. Hence we can categorize computing devices/computers into two types: ***mechanical*** and ***electronic*** computing devices/computers.



Use the Internet to perform these activities.

Activity 4.1 The real beginnings of computers as we know them today lay with an English mathematics professor, **Charles Babbage.** In 1822, what machine did he invent to be able to compute tables of numbers, such as logarithm tables? What did it use to make the machine run?

Difference engine

Activity 4.2 An important step in the history of computers was the design of a ***mechanical general-purpose computer*** by **Charles Babbage** in 1837. What machine did he invent?

Analytical engine

Activity 4.3 Who created a program for the mechanical general-purpose built by **Charles Babbage** in 1837.

Ada Lovelace

Activity 4.4 What is the ***Turing Machine***? Who developed it and in what year?

Turing machine is a mathematical model of computation that defines an abstract machine, which manipulates symbols on a strip of tape according to a table of rules. The Turing machine was invented in 1936 by Alan Turing

Activity 4.5 Who proposed the so called ***Stored-Program Computer*** concept or model in his paper that we are still using till these days? In what year did he propose?

John von Neumann in the late 1940s

Activity 4.6 What is the first electronic computer? Who created it and in what year?

Atanasoff - Berry computer created by John Vincent Atanasoff in 1930s

Activity 4.7 What is the first general-purpose electronic computer? Who created it and in what year?

ENIAC, John Mauchly and J. Presper Eckert

Activity 4.8 The history of computer development is often referred to in reference to the different generations of computing devices. Each generation of computer is characterized by a major technological development that fundamentally changed the way computers operate, resulting in increasingly smaller, cheaper, more powerful and more efficient and reliable devices. For example, the ***First Generation*** covers 1940-1956 and ***vacuum tubes*** were the fundamental change. Complete the following table:

	From Year	To Year	Fundamental Change
Second Generation	1956	1964	Transistors
Third Generation	1965	1975	Electronic computer Microprocessor
Fourth Generation	1975	Present	Microprocessor

QUESTIONS TO ANSWER:

1. What are the advantages of Computers?

Computers can increase a productivity of a people in our society because it helps in many fields like bussiness, education and many others. They have ability to store and process data allowing for the digitization of things like photographs, music, movies and books. They provides works to many people and connected us for giving information and to become sociable by using social media.

2. What are the disadvantages of Computers?

Computers can do a lot of people's what to do. People will become lazy because they depend on computers. They affect their health because they don't exercise regularly. Computers have many games that why people have addiction on them.

REFLECTIONS:

Computers have a lot of advantages but also have a disadvantages that can be manage. We use computers in a correct way that people use them to make their life compatible. They have a power to give improvements that can we use to make our country modernized.

Republic of the Philippines
POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
COLLEGE OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

CMPE 30022

Programming Logic and Design

LAB EXERCISE 3

GETTING STARTED WITH VISUAL LOGIC

Submitted by:
Robin Rosalosdos

Submitted to:
Engr. Julius Cansino

Date Submitted:
July 19, 2019

LAB 3: GETTING STARTED WITH VISUAL LOGIC

Duration: 3 Hours

Objectives

This lab activity aims student to:

1. Develop a good algorithm.
2. Construct flowcharting diagram based on a defined algorithm.
3. Familiarized with the different symbols used in flowcharting.
4. Familiarized with the use of a software called Visual Logic.
5. Transform a flowchart diagram into its equivalent pseudocode.
6. Acquianted with how to design a working computer program.

Task 1: Theoretical Framework

In this task, we will learn about a process of creating a computer program.

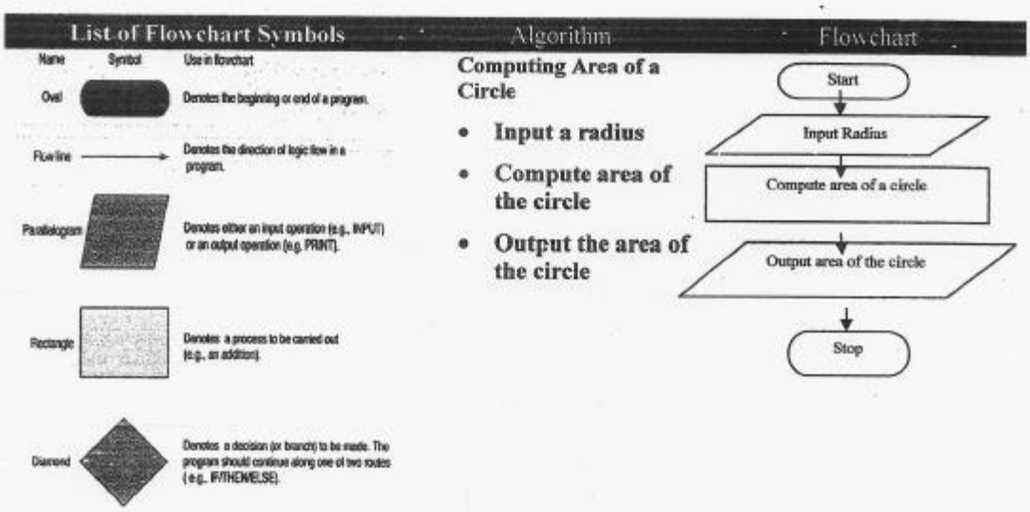
What is a computer programming?
A computer programming is a process of planning a sequence of instructions for a task or an event to be performed by a computer.

How do we write a program?
There are different phases to write a program -- Programming Life Cycle Phases:

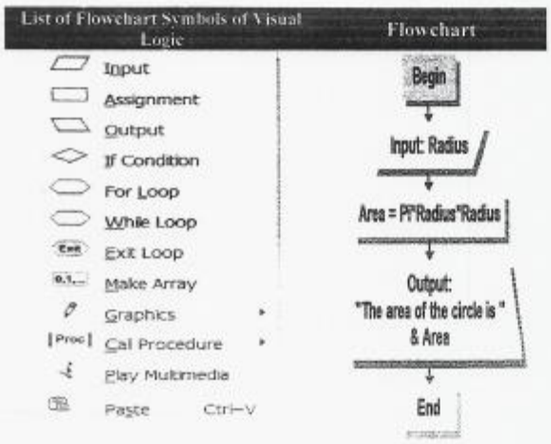
- Problem solving
  - *Analysis and Specification*: Determine precise objective of the solution to the problem
  - *Design a solution* (*Algorithm*): Develop a logical sequence of steps to solve the problem.
  - *Verification*: Check whether the solution does solve the problem
- Implementation
  - *Coding* (*Program*): Translate the design or algorithm into a programming language
  - With C++, you use
  - *Documentation* -- your written comments
  - *Compiler* -- translates your program into machine language
  - *Main Program* -- may call sub-algorithms
  - *Testing*: Have the computer follow the instruction in the program (*Run the program*) and check the results
  - If it does not, then you must find out what is wrong with your program or algorithm and fix it--this is called *debugging*
- Maintenance
  - *Utilization*: Use the program
  - *Maintain*: Revise or modify the program according to changing requirements

What is an algorithm?
An algorithm is a step-by-step procedure for solving a problem with a finite amount of data in a finite amount of time.

One way to represent an algorithm is to use flowchart. The following is a list of flowchart symbols that are used to describe an algorithm.



We will use software called **Visual Logic** to create flowcharts for our programs that we will develop in class. Here is how the flowchart symbol of **Visual Logic** looks like:



Activity 1.1: Use Visual Logic to create a flowchart of an algorithm to find the area of a circle with a given radius as shown in the above picture.

Activity 1.2: Run the program (flowchart) in Activity 1.1. Did it work?

Yes.

Activity 1.3: Use **Visual Logic** to create a flowchart for the algorithm in Activity1.1. Then run to see if the algorithm correct.

Task 2: Move on Screen

Activity 2.1: Use **Visual Logic** to create a flowchart for each algorithm. Then run to see if the algorithm correct. You are to cut and paste to this document the following: the flowchart you've created from Visual Logic and the captured output screen.

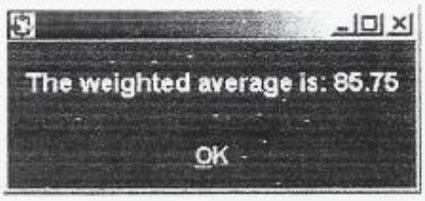
Create a flowchart using Visual Logic to represent the logic of a program that allows the user to enter two numbers and then calculate and display the sum, difference, quotient, product, and average of the two numbers. Enter 15 for the first number and 10 for the second number. Sample output is below:

Activity 2.2: Use **Visual Logic** to create a flowchart for each algorithm. Then run to see if the algorithm correct. You are to cut and paste to this document the following: the flowchart you've created from Visual Logic and the captured output screen.

Create a flowchart using Visual Logic to represent the logic of a program that calculates the weighted average of four exams. The respective weights and scores are given below:

Exam 1 20% 100
Exam 2 35% 85
Exam 3 15% 90
Exam 4 30% 75

The program should allow the user to enter the score for each exam from the keyboard, calculate the weighted average, and display it to the screen. Sample output is shown below:

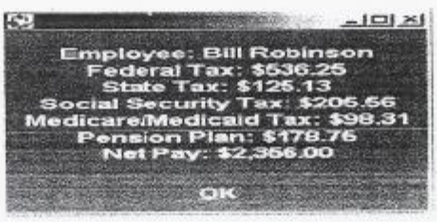


Activity 2.3: Use **Visual Logic** to create a flowchart for each algorithm. Then run to see if the algorithm correct. You are to cut and paste to this document the following: the flowchart you've created from Visual Logic and the captured output screen.

Create a flowchart using Visual Logic to represent the logic of a program that calculates and displays the monthly paycheck for an employee. The net pay is calculated after taking the following deductions:

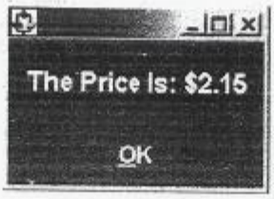
Federal Income Tax: 15%
State Tax: 3.5%
Social Security Tax: 5.75%
Medicare/Medicaid Tax: 2.75%
Pension Plan: 5%
Health Insurance: $75.00

The Health Insurance deduction is a fixed amount for the year. Your program should prompt the user to enter the employee name and gross amount from the keyboard. Use Bill Robinson for the employee name and $3575.00 for gross amount. Numeric output should be in currency format. Sample output is shown below:



Activity 2.4: Use **Visual Logic** to create a flowchart for each algorithm. Then run to see if the algorithm correct. You are to cut and paste to this document the following: the flowchart you've created from Visual Logic and the captured output screen.

Create a flowchart using Visual Logic to represent the logic of a program that will calculate the price of an item. The program should allow the user to enter the price of an item and whether or not that item is taxable as indicated by "Y" or "N" from the keyboard. If the item is taxable the program should calculate the tax based on the current tax rate of 8.25% and recalculate the pr ice, otherwise, nothing should be done. Finally, the program should display the price. For example if the user enters $1.99 for the price of the item and "Y" to designate the item is taxable, the program should produce the sample output shown below (price should be displayed in currency format):



If the user enters $1.99 for the price of the item and "N" to designate the item is not taxable, the program should produce the sample output shown below (price should be displayed in currency format):

QUESTIONS TO ANSWER:

1. What is algorithm?

Algorithm is a step by step method of solving used for data processing, calculation and other mathematical operations.

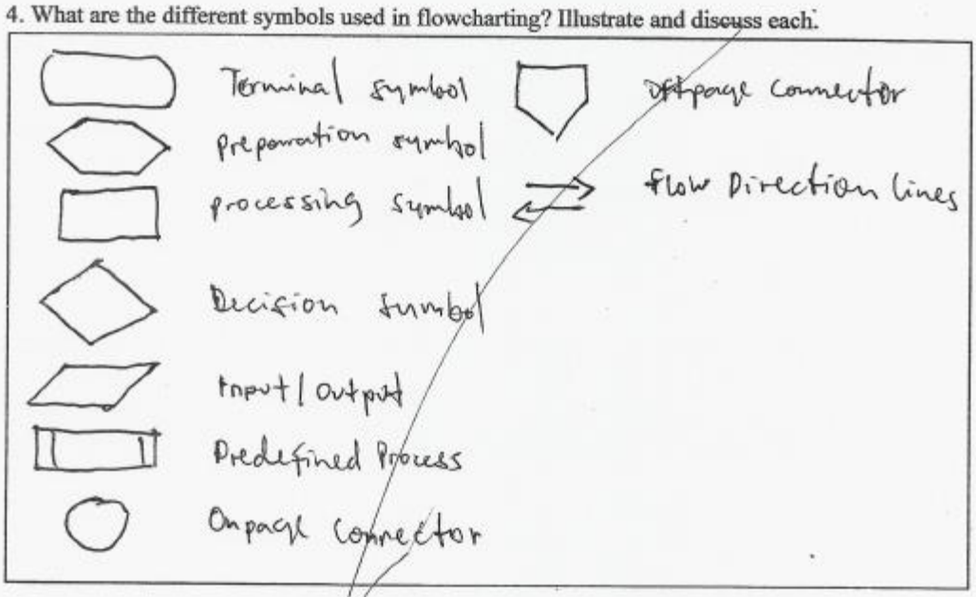
2. What is program logic?

Program logic is an instructions in a program arranged in a prescribed order to solve a problem.

3. What is flowcharting?

Flowcharting is creating a diagram that represents an algorithm, workflow or process. That called flowchart. It shows the steps as boxes of various kinds and their order by connecting the boxes with arrows.

4. What are the different symbols used in flowcharting? Illustrate and discuss each.



REFLECTIONS:

Flowcharting is the simplest way in order to manipulate logics, algorithm or a program. It is a graphic diagram that can easily understand by end-users. It have many symbols in order to create a flow of your program.

Republic of the Philippines
POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
COLLEGE OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

CMPE 30022
Programming Logic and Design

LAB EXERCISE # 3
GETTING STARTED WITH VISUAL LOGIC

SUBMITTED BY:

PASALOSDOS, Robin I.

BS Computer Engineering 1-4

SUBMITTED TO:

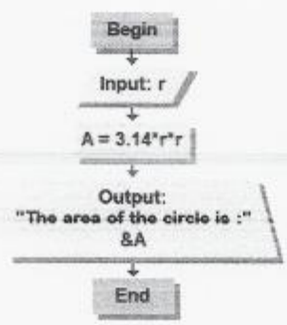
ENGR. JULIUS S. CANSINO

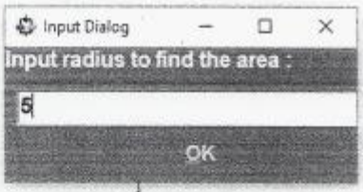
Instructor, Programming Logic and Design

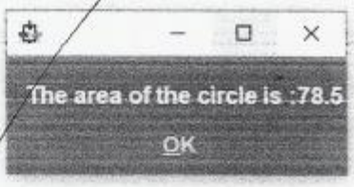
DATE SUBMITTED:

July 19, 2019

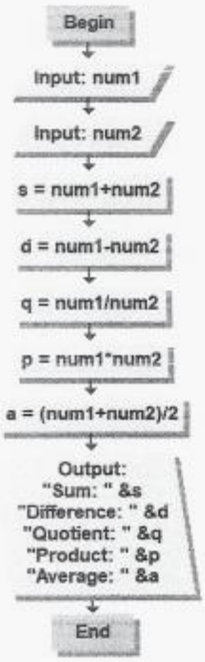
Act. 1.1

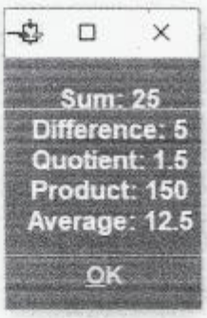

Begin
Input: r
A = 3.14*r*r
Output:
"The area of the circle is :"
&A
End


Input Dialog
Input radius to find the area :
5
OK

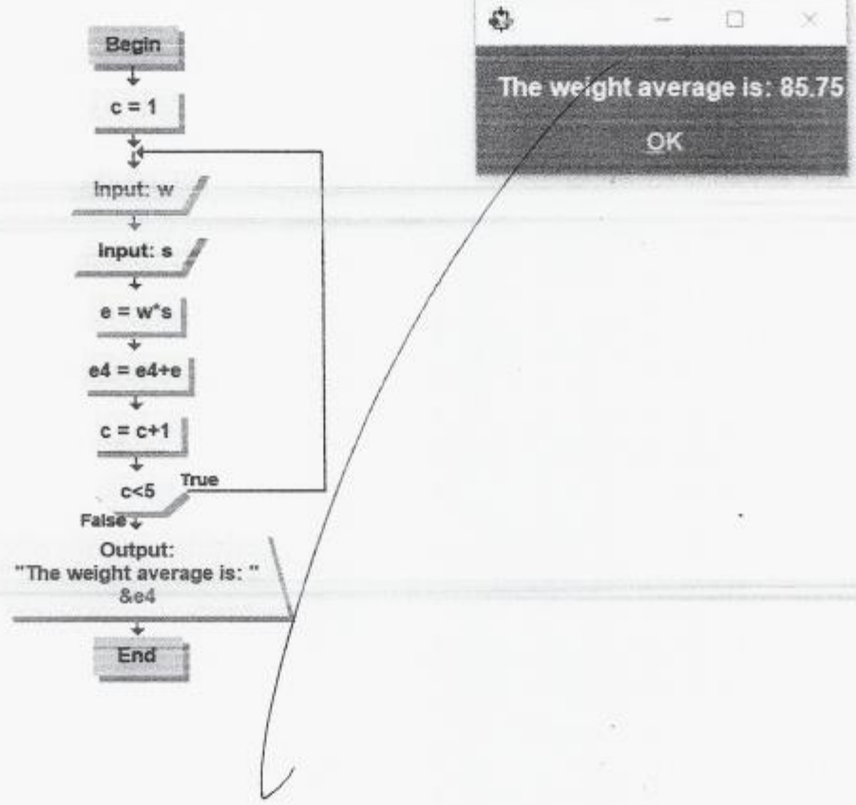

The area of the circle is :78.5
OK

Act. 2.1

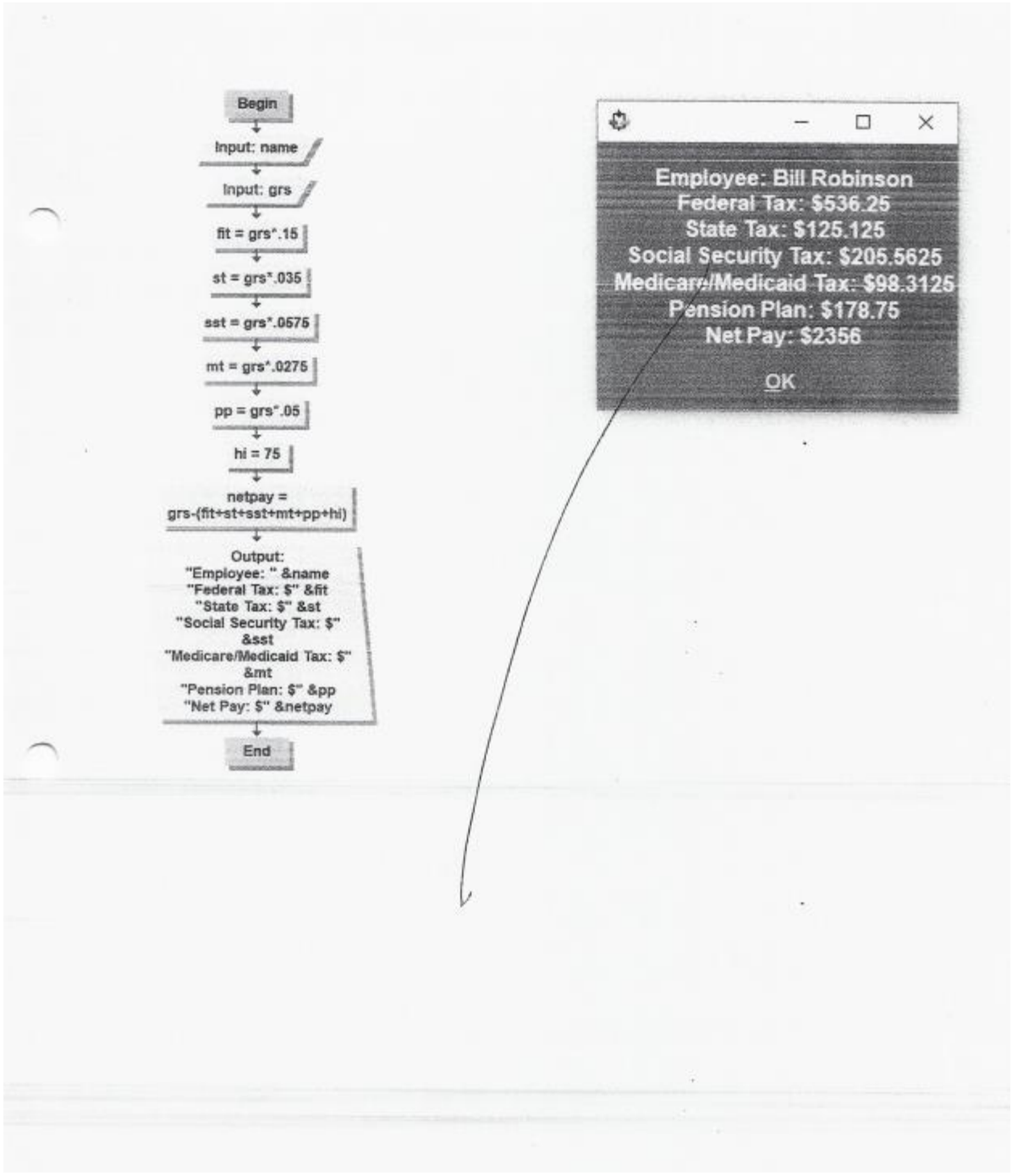

Begin
Input: num1
Input: num2
s = num1+num2
d = num1-num2
q = num1/num2
p = num1*num2
a = (num1+num2)/2
Output:
"Sum: " &s
"Difference: " &d
"Quotient: " &q
"Product: " &p
"Average: " &a
End


Sum: 25
Difference: 5
Quotient: 1.5
Product: 150
Average: 12.5
OK

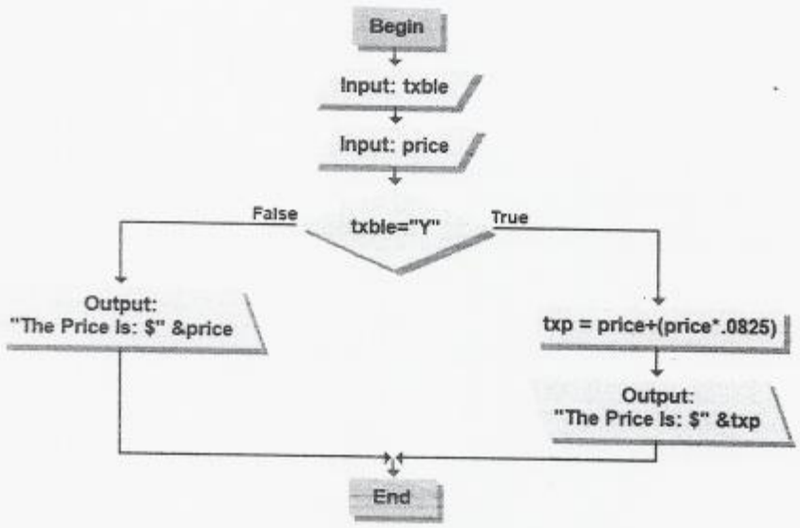
Act. 2.2

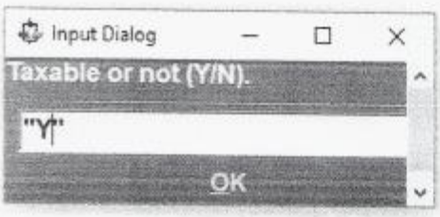

Begin
c = 1
Input: w
Input: s
e = w*s
e4 = e4+e
c = c+1
c<5
True
False
Output:
"The weight average is: "
&e4
End
The weight average is: 85.75
OK

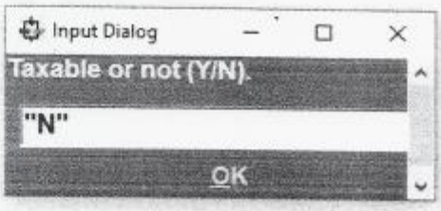
Act. 2.3

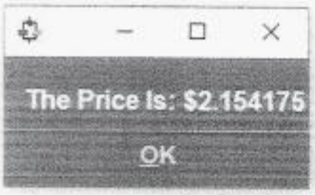

Begin
Input: name
Input: grs
fit = grs*.15
st = grs*.035
sst = grs*.0575
mt = grs*.0275
pp = grs*.05
hi = 75
netpay = grs-(fit+st+sst+mt+pp+hi)
Output: "Employee: " &name "Federal Tax: $" &fit "State Tax: $" &st "Social Security Tax: $" &sst "Medicare/Medicaid Tax: $" &mt "Pension Plan: $" &pp "Net Pay: $" &netpay
End
Employee: Bill Robinson
Federal Tax: $536.25
State Tax: $125.125
Social Security Tax: $205.5625
Medicare/Medicaid Tax: $98.3125
Pension Plan: $178.75
Net Pay: $2356
OK

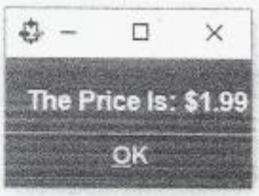
Act. 2.4


Begin
Input: txble
Input: price
txble="Y"
False
True
Output:
"The Price Is: $" &price
txp = price+(price*.0825)
Output:
"The Price Is: $" &txp
End


Input Dialog
Taxable or not (Y/N).
"Y"
OK


Input Dialog
Taxable or not (Y/N).
"N"
OK


The Price Is: $2.154175
OK


The Price Is: $1.99
OK

Republic of the Philippines
POLYTECHNIC UNIVERSITY OF THE PHILIPPINES
COLLEGE OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

CMPE 30022

Programming Logic and Design

LAB EXERCISE 2

COMPUTER NUMBERING SYSTEM

Submitted by:
Robin Pasalosdos

Submitted to:
Engr. Julius Cansino

Date Submitted:
July 5, 2019

LAB 2: Computer Numbering System

Duration: 3 Hours

Objectives:

- Learn how decimal numbers or numbers in base 10 are represented in other bases.
- Understand how data is organized.
- Understand the units use to measure the size of information and storage.
- Learn how to convert numbers in base 10 to other bases.
- Learn how unsigned (positive and zero) decimal numbers are represented in binary.

Introduction

Before we focus on information representation, we need to know what a computer is and what it does. So, what is a ***computer***? From:

1. The American Heritage® Dictionary: a ***computer*** *is a device that computes, especially a programmable electronic machine that performs high-speed mathematical or logical operations or that assembles,* ***stores****, correlates, or otherwise* ***processes information****.*
2. The Columbia Encyclopedia: a ***computer*** *is a device capable of performing a series of arithmetic or logical operations. A* ***computer*** *is distinguished from a calculating machine, such as an electronic calculator, by being able to* ***store*** *a computer program (so that it can repeat its operations and make logical decisions), by the number and complexity of the operations it can perform, and by its ability to* ***process****,* ***store****, and retrieve* ***data*** *without human intervention.*

In short, a computer is a device capable of ***storing***, ***retrieving***, and ***processing information*** or ***data***. What types of information can a computer store/process? Information can roughly be categorized into 5 groups:

1. Numbers
2. Characters
3. Pictures
4. Sounds
5. Instructions or Computer Programs

This leads to a question how numbers, characters, pictures, sounds, or instructions can be ***represented*** such that they can be stored and processed by a computer. The answer is ***binary numbers***. Therefore we will focus on ***information representation*** in a computer. Let us first explore how numbers information can be represented as binary numbers.

The number system that we accustom to and use in our everyday lives is decimal number system or base 10. On the other hand computers, by designed, store information as a sequence of 0s and 1s. The number system contains only 0 and 1 is called ***binary number system*** or base 2. So it is necessary for us to know how decimal numbers are represented in a computer system using binary numbers. Before we find out how to represent decimal numbers using binary numbers, it is a good idea to explore numbers in different bases and understand how numbers in different bases are formed.

Task 1: Base Number

As previously mentioned, we accustom to decimal numbers (base 10) but computers are designed to use binary numbers (base 2). In decimal or base 10 there are 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Activity 1.1: How many digits are in *binary* or base 2? What are the digits?

Their have a 2 digits, The digits are 0 & 1.

Activity 1.2: How many digits are in *hexadecimal* or base 16? What are the digits? (Use A for 10, B for 11, and so on.)

Their are 16 digits, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F.

Activity 1.3: In base 10 the next ten numbers after 9 are 10_{10}, 11_{10}, 12_{10}, 13_{10}, 14_{10}, 15_{10}, 16_{10}, 17_{10}, 18_{10}, and 19_{10}. Subscription 10, $\square_{10}$, indicates the number is decimal or in base 10. They are obtained from the combination of 1 and the digits 0 to 9. What are the next two numbers in base 2 after 1_2? **Observe that we use a subscription to denote the base that the number is in.**

10_2 and 11_2

Activity 1.4: What are the next two numbers in base 16 after F_{16} (15_{10})?

10_{16} and 11_{16}

Activity 1.5: What is the next number in base 2 after 011_2?

100_2

Activity 1.6: What is the next number in base 16 after $0FF_{16}$?

100_{16}

Activity 1.7: Are the value 10_{10} (10 in base 10), and 10_2 (10 in base 2) the same? Why or why not?

No, because 10_2 is equal to 2_{10}, $10_{10} \neq 2_{10}$.

Task 2: Data Organization

Computers store information as binary numbers, that is, a sequence of 0s and 1s. Since computers have limited size of storage, computer engineers need to have a way to measure the size of information. The following terminology is used to measure information that represented in computers.

- *Bits* stand for **B**inary digits. A ***bit*** is the fundamental unit of computer storage. Related bits are grouped to represent different types of information such as numbers, characters, pictures, sound, and instructions
- *Nibbles* – A ***nibble*** is a group of four bits. It is the size of a hexadecimal digit.
- *Bytes* – A ***byte*** is a group of 8 bits.
- *Words* – A ***word*** is a group of 16 bits or 2 bytes.

Activity 2.1: How many different combinations or numbers that 0s and 1s with 2 bits can form? (Hint: List all the possible combinations that binary numbers with 2 bits can form.)

There are 4 combinations 00,01,10 and 11

Activity 2.2: How many different combinations or numbers that 0s and 1s with 3 bits can form?

8 combinations.

Activity 2.3: How many different combinations or numbers that 0s and 1s with 4 bits can form?

16 combinations.

Activity 2.4: In general, how many different combinations or numbers that 0s and 1s with *n* bits can form? That is to find a formula to determine how many numbers that *n* bits can form.

2^n

Activity 2.5: The size of storages or memory units of computers is usually measure in bytes. These units vary upon the size of storages. They can be measured as *kilobyte* (KB), *megabyte* (MB), *gigabyte* (GB), *terabyte* (TB) and *petabyte* (PB). A *kilobyte* is roughly one thousand bytes. But the actual number is 1,024 that is 2^{10}. That means 10 bits can form exactly 1,024 binary numbers. We use kilobyte refer to 1,024 bytes because it is easy to reference. A megabyte is roughly one million bytes. What is the actual number of bytes in a megabyte? (Feel free to use any computation device.)

1048576 bytes

Activity 2.6: What is a gigabyte?

1073741824 bytes

Activity 2.7: What is a terabyte?

$1.099511628 \times 10^{12}$ bytes

Task 3: Decimal Numbers

Let us first look at how to convert an unsigned or nonnegative binary number to a decimal number. If you recall from what you learned in elementary school how the position of each digit in a decimal number affects the value of the digit, this is called *positional notation*. For example,

$$2401_{10} = 2\times10^3 + 4\times10^2 + 0\times10^1 + 1\times10^0$$

Note that decimal numbers are number in base 10. *We always start counting position from 0 and from right to left*. Since 4 is in position 2, it has the value 400 which is 4 times 10 to the power of 2, that is 4×10^2. Similarly, since 1 is in position 0, it has the value 1 which is 1 times 10 to the power of 0. You can see that 10 refers to the base of the number 2401 that it represents, and 2 and 0 refer to the positions of digit 4 and 1, respectively.

An easier way to understand this is to view the position of digits and the corresponding positional values as a table as follows:

Position	...	4	3	2	1	0
Positional Value	...	10^4	10^3	10^2	10^1	10^0
	...	10000	1000	100	10	1
			2	4	0	1

$$2\times1000 + 4\times100 + 0\times10 + 1\times1 = 2401_{10}$$

Fortunately, we can use positional notation as the above example to convert binary numbers to decimal number or number in base 10.

For example, we want to convert a binary number 10110_2 to decimal. We, first, create a table that indicates positional values for base 2 to organize our conversion process by writing down the values of each position in ascending order from right to left. For example, in base 2, the positional values are: `... 256 128 64 32 16 8 4 2 1`. These values are obtained as follows:

Position	...	8	7	6	5	4	3	2	1	0
Positional Value	...	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	...	256	128	64	32	16	8	4	2	1

So, the table of positional values for base 2 looks as follows:

...	256	128	64	32	16	8	4	2	1

Then place each digit or bit of the given number from right to left. So, the table should look as follows:

...	256	128	64	32	16	8	4	2	1
					1	0	1	1	0

The table indicates the conversion of 10110_2 to decimal is

$$1\times16 + 0\times8 + 1\times4 + 1\times2 + 0\times1 = 16 + 4 + 2 = 22_{10}$$

Therefore $10110_2 = 22_{10}$

Note that the converted decimal number is the sum of all positional values that have a corresponding 1 under them.

So, to convert a binary number to decimal number using position notation can be done as follows:

1) List enough positional values for base 2 from right to left.
2) Place each bit of a given binary number under each positional value from right to left.
3) The decimal number is the sum of all positional values that have a corresponding 1 under them.

Activity 3.1: What decimal number does binary number 11010_2 represent?

26_{10}

Activity 3.2: What decimal number does binary number 100111_2 represent?

39_{10}

Task 4: Unsigned Binary Numbers

In Task 3 we use the positional notation to convert unsigned binary numbers to decimal numbers. In this task our focus is on representation of unsigned or nonnegative decimal numbers (integers) by binary numbers. In mathematics, a set of integers consists of negative integers, zero, and positive integers. So, in computer systems, there is a need to represent negative integers and positive integers including zero. Therefore we divide integers into *unsigned* and *signed* numbers. Unsigned numbers are only zero and positive integers. On the other hand, signed numbers are negative integers, zero, and positive integers or simply integers. So, we need number systems to represent these two groups of integers. There are four number systems that are used to represent integers: unsigned binary, signed and magnitude, 1's complement and 2's complement number systems. We will focus on only unsigned binary. **Note that unsigned binary is used to represent only positive integers and zero.**

There are two ways to convert a decimal number to a binary number.

1. Successive division by 2: Divide the given decimal number by 2 successively and collect the remainders in backward order. The process will be terminated if the quotient is 0.

   For example: What is representation of 79_{10} in binary?

2 | 79		
2 | 39		1
2 | 19		1
2 | 9		1
2 | 4		1
2 | 2		0
2 | 1		0
0		1

   Therefore $79_{10} = 1001111_2$

2. Reverse positional notation: Use positional notation in reverse, that is, make use of the value in each position ($2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$,...) to determine which position is a 1. The following is the process of conversion:
    1) Write down or create a table of the values of each position in ascending order from right to left. For example: ... `256 128 64 32 16 8 4 2 1` Make sure that the value of the left most number is larger than the decimal number to be converted. Note that the value `1` is 2^0, `2` is 2^1, `4` is 2^2, `8` is 2^3,...
    2) Determine which position that has the largest value but less than or equal to the given decimal number and mark that position as a 1. For example, the number to be converted is `79`. Then the position `64` will be marked as a 1 since `79` is less than `128` but greater than `32`.
    3) Subtract that positional value from the given decimal number to obtain the remaining value for the next step. From the example in step 2, the positional value is `64`. Then the remaining value is `79 – 64 = 15`
    4) Repeat the process with the value from the step 2. That means the position `8` will be marked as a `1`. Since `15` is less than `16` but greater than `8`, and the remaining value is `15 – 8 = 7`
    5) The positions that are skipped will be mark as a 0. That means the position `32` and `16` will be marked as 0.
    6) The process will be terminated if the value in the third step is 0.

So, the number $79_{10} = 1001111_2$

64	32	16	8	4	2	1
1	0	0	1	1	1	1

Activity 4.1: Convert the integer 209_{10} to an unsigned binary using the reversed-positional notation.

11010001_2

Activity 4.2: In a lot of cases, it is more convenient for programmers to deal with information in hexadecimal representations rather than binary representations. That means the programmers should know how to convert between binary and hexadecimal representation. Fortunately, the process of conversion from binary to hexadecimal and vice versa is relatively simple.

- To convert a binary representation to hexadecimal:
    1. Divide binary representation from right to left in to group of 4 bits (nibble).
    2. If the last group has less than 4 bits, pad 0s (to left) to make up 4 bits.
    3. Convert each group of 4 bits to a corresponding hexadecimal digit.

  For example, $1110100111011_2 = 0001\ 1101\ 0011\ 1011_2 = 1D3B_{16}$
- To convert a hexadecimal representation to binary, just covert each hex digit to corresponding binary representation.

  For example, $9C3F_{16} = 1001\ 1100\ 0011\ 1111_2$

The following table relates binary, decimal, and hexadecimal:

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

What is the Hex representation of the binary number 1100101100_2?

$32C_{16}$

Activity 4.3: What is the binary representation of the hexadecimal number $2A8DB9_{16}$?

10101010001101101110012

QUESTIONS TO ANSWERS:

1. What is the next number after 101_2 in base 2?

110_2

2. What is the next number after 99_{10} in base10?

100_{10}

3. List all binary numbers from 0 to 7.

$0_2, 1_2, 10_2, 11_2, 100_2, 101_2, 110_2, 111_2$

4. What is the actual number of bytes in 1 Kilo byte?

1024 bytes

5. How many bits does it require to represent binary from 0 to 15? List all binary numbers.

4 bits; $0000_2, 0001_2, 0010_2, 0011_2, 0100_2, 0101_2, 0110_2, 0111_2, 1000_2$
$1001_2, 1010_2, 1011_2, 1100_2, 1101_2, 1110_2, 1111_2$

6. What is the decimal value of binary number 100_2?

4_{10}

7. What is the decimal value of binary number 1101_2?

13_{10}

8. Convert the integer 23_{10} to a binary.

10111_2

9. Convert the integer 113_{10} to a binary.

1110001$_2$

10. Convert the integer 178_{10} to a binary.

10110010$_2$

REFLECTIONS:

Computer can only understand the two digits, 0 and 1. People use 10 digits for their guides like counting, solving mathematical equation etc. The number system of computer and people are different to each other so that we can use conversions. The hexagonal and octal numbers were developed because binary numbers in computers can easily understand by humans. We find many ways for better relationship to computers.