Week 1 Agenda

Introduction to programming concepts

- Types of programming languages
- Solving problems
- How computers work and binary introduction
- Solving problems
- Algorithms
- Flowcharts/pseudocode

Computer

A computer is an electronic device which takes input from the user, processes the input, and outputs the results for the user.

Computers include: cell phones, iPods, and Tablets.

Input consists of any form of instruction or data the user provides the computer.

Process consists of processing instructions and data and storing results.

Output consists of displaying the stored results or printing the output.

Computer Hardware

A computer system consists of multiple pieces of hardware that allow them to function.

Central Processing Unit (CPU) - The brain of the machine. This is where all the basic operations of the computer are carried out. The CPU tells the rest of the computer what to do.

Memory - The memory of a machine stores programs and data. Computers have two different types of memory: main and secondary memory.

Input/Output devices - Input and output devices all users to interact with the computer. Examples of Input devices are the computer mouse and keyboard. Examples of output devices are the computer's monitor. Information from input devices is process by the CPU and may be stored in main memory or secondary memory such as RAM.

Main Memory

Considered the computer's work area

Computer stores the program that is running as well as the data

Commonly known as the ***random-access memory (RAM)***

Data is quickly accessed

RAM is a volatile type of memory

Used for temporary storage

RAM is erased when computer is turned off

Secondary Memory

Type of memory that can hold data for long periods of time.

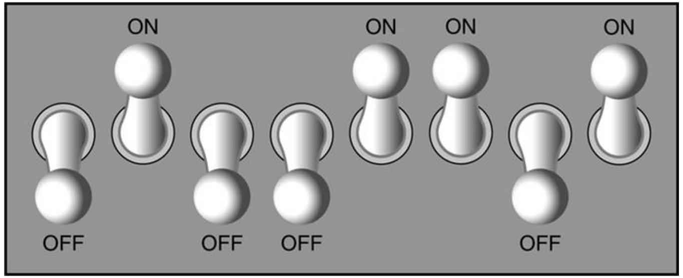
Programs and important data are stored in secondary storage.

Disk drive is a common type of secondary storage

- Data is stored by magnetically encoding it onto a circular disk
- Most computers have an internal disk drive
- Some have external disk drives; they are used to create backup copies

Storing Information

- A computer’s memory is divided into tiny storage locations known as ***bytes.***
- One byte represents one number.
- A byte is divided into eight smaller storage locations known as ***bits (binary digits).***
- Bits are tiny electrical components that can hold either a positive or a negative charge.
- A positive charge is similar to a switch in the *on* position.
- A negative charge is similar to a switch in the *off* position.



Storing Numbers

- The positive charge or the *on* position is represented by the digit 1
- The negative charge or the *off* position is represented by the digit 0
- This corresponds to the binary numbering system where all numeric values are written as a sequence of 0s and 1s
- Each digit in a binary number has a value assigned to it

For more information on binary visit: https://www.codeproject.com/Articles/4069/Learning-Binary-and-Hexadecimal

Computer Program/Software

A computer program is a sequence of instructions that are processed by a computer's CPU and tell the computer to perform a specific set of tasks. These instructions must be in a language that the computer can understand (machine code). In order to get these instructions into machine code a software developer writes a source code in the form of a program using one of many programming languages such as Python. When source code is written it must follow rules specific to the language called syntax.

Two categories of software

- System software
- Application software

Compiled and Interpreted Programs

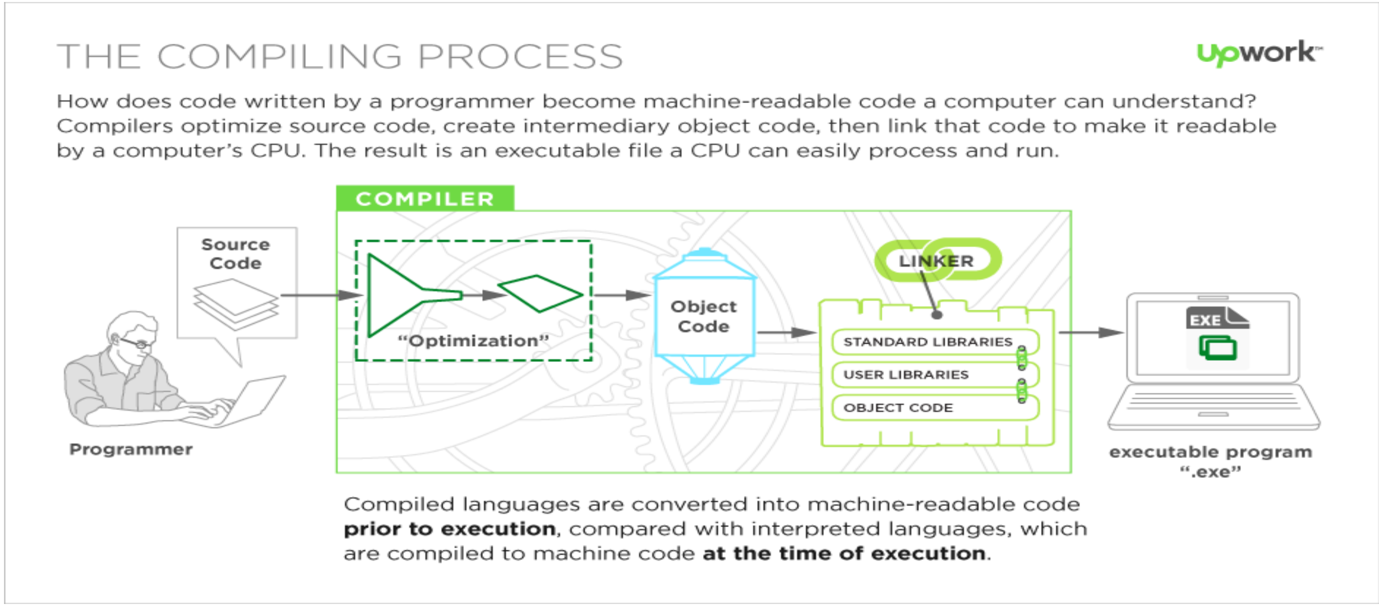
Compiled Languages - Compiled languages require a compiler in order to convert source code into a computer's native language (machine code). Once the code is compiled the application can be executed on the chosen platform. With compiled languages the source code only needs to be compiled once and then the application can be run multiple times. In most cases compiled languages execute faster than interpreted languages.

C, C++, Java, ASP.NET

Interpreted Languages - Interpreted languages use an interpreter to convert source code into machine code one line at a time when the application is executed. Each time the program is run the source code is interpreted.

Python, Perl, Javascript

Compiling Process



THE COMPILING PROCESS

How does code written by a programmer become machine-readable code a computer can understand? Compilers optimize source code, create intermediary object code, then link that code to make it readable by a computer's CPU. The result is an executable file a CPU can easily process and run.

Compiled languages are converted into machine-readable code **prior to execution**, compared with interpreted languages, which are compiled to machine code **at the time of execution**.

Algorithm

In order to solve a problem computationally, two things are needed: a representation that captures all the relevant aspects of the problem, and an algorithm that solves the problem by using of the representation.

An algorithm is a series of steps that can be systematically followed for producing the answer to a certain type of problem.

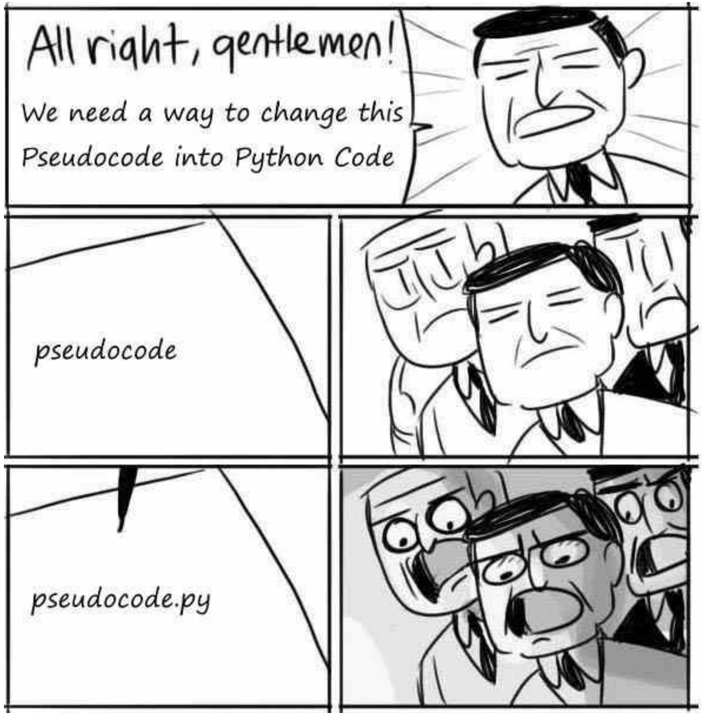
Software Development Process

1. The first step in programming is designing – **flowcharts** and **pseudocode** help with this process.
2. Next, the code is written.
3. All code must be cleared of all **syntax errors**.
4. After the executable is created, it can be checked for **logic errors.**
5. If logic errors exist, the program must be **debugged**.

Program Design

Two steps in designing a program

1. Understand the tasks that the program is to perform.
    a. Learning what the customer wants.
2. Determine the steps that must be taken to perform the task.
    a. Create an algorithm, or step-by-step directions to solve the problem.
    b. Use flowcharts and/or pseudocode to solve.


All right, gentlemen!
We need a way to change this Pseudocode into Python Code
pseudocode
pseudocode.py

Pseudocode

Fake code used as a model for programs

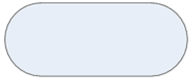
No syntax rules

Well written pseudocode can be easily translated to actual code

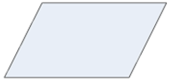
```
Display “Enter the number of hours”
Input hours
Display “Enter the hourly pay rate”
Input payRate
Set grossPay = hours * payRate
Display “The gross pay is $”, grossPay
```

Flowcharts

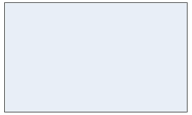
A diagram that graphically depicts the steps that take place in a program



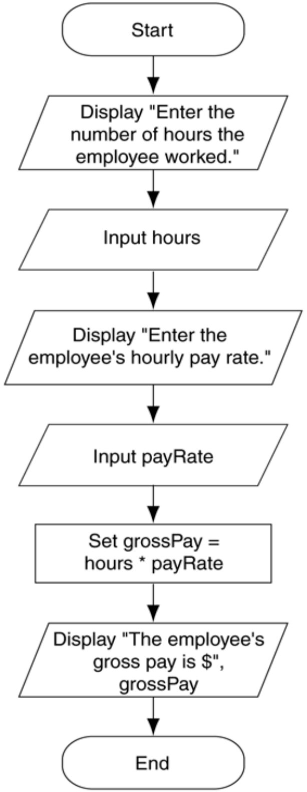
Terminator used for start and stop



Parallelogram used for input and output



Rectangle used for processes



Start

Display "Enter the number of hours the employee worked."

Input hours

Display "Enter the employee's hourly pay rate."

Input payRate

Set grossPay = hours * payRate

Display "The employee's gross pay is $", grossPay

End

References

The Basics of Compiled Languages, Interpreted Languages, and Just-in-Time Compilers, https://www.upwork.com/hiring/development/the-basics-of-compiled-languages-interpreted-languages-and-just-in-time-compilers/

The End

You may close this Window and return to the course.