Computer Science 110
Syllabus

Print the Syllabus for future reference.

Course Description:
Computer Science 110 covers the fundamentals of computer programming. You will learn to program with Visual Basic 6.0. No prior programming experience is necessary to take the course.

Teacher:

Mr.Hopper

Email: luke.hopper@nbed.nb.ca

Getting Started:
You will need the following to get started in the course:

- Your user ID and password (which you must have by now if you are logged into this course)
- The course URL https://nbvhs.nbed.nb.ca bookmarked on your computer
- Visual Basic 6.0 loaded on your computer
- Microsoft Word and PowerPoint loaded on your computer

The course website contains all the information you need to learn to program with Visual Basic. There is no textbook to accompany the course. The website is your online textbook. Once you have read through all the introductory information in the syllabus, you are ready to start.

Moving Through the Course:

- There are 19 units of work to be completed. Begin with unit one and do each unit, in order, until you are finished.
- At the end of most units, there is a multiple choice quiz and an assignment that must be completed before proceeding to the next unit.
- A PowerPoint presentation at the end of each unit will help you review the unit content before you take your quiz.

Submitting Assignments:
All assignments are done in Visual Basic or with Microsoft PowerPoint or Word. Create a folder on your hard drive to save your work. For each assignment, you will save your file with a specific file name. Make sure you name your files as instructed. You will submit your assignment files by uploading them through the Dropbox link.

Evaluation: Your final mark for the course will be calculated as shown:

Quizzes	35%
Assignments	35%
Final Exam	30%

Students will receive a mark of zero (0) for any work that is not completed. The final exam is released when the last assignment is submitted and graded.

Note: No mid-semester mark is given on the student's report card. The course is considered to be "in progress."

Planning Your Time: The semester is 18 weeks long. There are 19 units in the course. You should be able to move through the first five units very quickly. Plan to be finished to the end of Unit 5 after two weeks. After that, plan to spend approximately one week on each unit, including time for the unit assignment and the quiz. You will most likely have to spend time outside of class time to maintain this schedule. Be sure to allow yourself time at the end to review before your final exam.

Cheating: Cheating on assignments is not allowed. The following are examples of cheating:

- Copying someone else's assignment
- Taking someone else's assignment, making a few minor changes such as changing variable names and then submitting it as your own work
- Having someone else do the assignment for you

A mark of 0 will be given to each student who is involved in cheating. I.e. the student who copied the assignment will receive zero (0) and the student who owned the assignment that was copied will receive zero (0). No exceptions to this will be granted. If caught for cheating, you could be asked to withdraw from the course.

I print and save a copy of every assignment. It is very easy to spot a copied assignment!

Guidelines for Using the Chat Room: The chat room is not a public chat room. It is not meant for socializing with other students. The chat room that is part of this course is your **online classroom**. The expectations that I have of you in the chat room are no different than I would expect if we were face-to-face in a classroom.

These expectations are:

- Use good manners.
- Use appropriate language.
- Be patient if the room is "busy." You may have to wait for your turn if there are a lot of students who have questions. Sometimes sending me an email is a better way to get the help you need!

Unit 1: Computer Systems
Unit 1 Objectives

By the end of this unit, you should be able to:

- Identify the components of a computer system.
- Identify input and output devices.
- Explain the importance of the computer's processor.
- Explain how high level programming languages work.
- Explain why software portability is desirable.
- Distinguish between main and secondary memory.
- Distinguish between system software and application software.

Components of a Computer System

The term **computer** refers to a system that is made up of many components, or parts.

Computer systems are made up of:

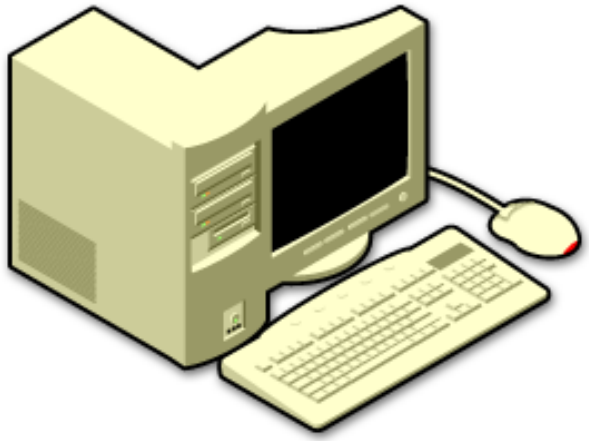
- Hardware components —the hardware components are made up of the electronic and mechanical parts.
- Software components —the software components are made up of the computer programs and the data.

The software components are stored on the computer's hardware components.

Hardware Components

The major hardware components of a computer system are:

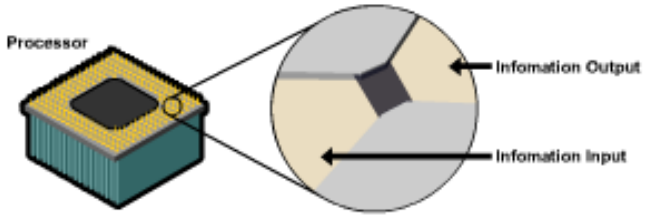
- Processor
- Main memory
- Secondary memory
- Input devices
- Output devices



The processor, main memory and secondary memory are inside the system unit. The system unit is the metal box that we refer to as **the computer**.

Your monitor, keyboard and mouse are outside the system unit. The monitor is considered an output device and the keyboard and mouse are considered input devices.

Input and output operations are so common that an abbreviation has been created for them: **I/O** (pronounced eye-oh). I/O operations are very complicated at both the hardware and software levels. The reason they are so complicated is because the way in which data is organized **outside** the computer is different from the way it is organized **inside** the computer.

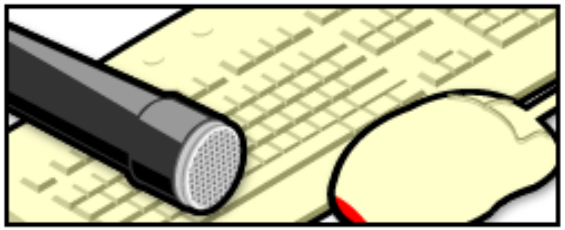


In the picture, each red dot represents a piece of data waiting its turn to be input. Each blue square is the processed information leaving as output. Inputting data is usually referred to as **reading** data and outputting data is referred to as **writing** data. The information that comes out as output is not usually the same as the data that was input. It has been processed inside the computer system.

Input and Output Devices

Every component of your computer system that is outside the processor is either an input device or and output device.

Input Device —an input device is used to bring data into the computer system.



Input devices include:

- Keyboard
- Mouse
- Microphone

Output Device —an output device is used to send information out of the computer system.

Output devices include:

- Monitor
- Printer
- Speakers

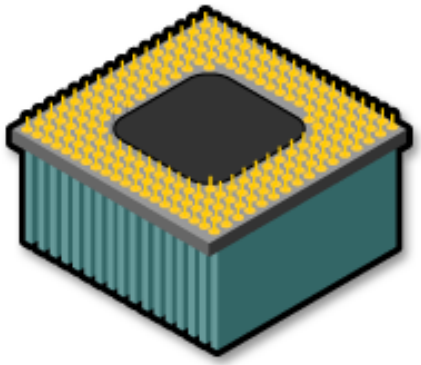
Input and output devices are usually called I/O devices. Each device is directly connected to a device controller inside the system unit. For example, your computer's speakers are directly connected to an audio card such as Sound Blaster. The audio card is connected to the rest of your system.

The Computer's Processor

Inside the computer box you will find the **processor**, sometimes called the **CPU** or **central processing unit**. The processor is a small, electronic device that is about 2.5 cm square.

The processor is the **brain** of the computer system. All basic computing is carried out by the CPU. The CPU also directly or indirectly controls all the other parts of the computer system.

Different types of computers have different types of processors. A Pentium processor is the type of processor that is probably inside the computer you are working on right now.



As a computer program runs, the processor constantly performs many tiny electronic operations. These operations might include reading a byte of data from main memory into the processor or testing one of the bits in a byte to see if it is a **1** bit. Today's processors can perform several hundred types of small operations.

These types of operations are the only things that a processor can do. Also, everything that a processor does is built out of tiny operations like those described above. The operations are performed one at a time.

Fortunately for us, with today's **high-level languages**, we do not need to know the details of these operations to write programs.

The processor performs **machine operations**, one after the other. Several thousands of operations can occur every second.

Machine instructions tell the processor which operations to perform and when to perform them.

The processor can only understand machine language instructions. A **machine language program**, more commonly referred to as an **executable program**, is a collection of machine instructions. These instructions are held in main memory and executed, one at a time, by the processor as the program runs, or executes.

To summarize:

- A machine language program is a sequence of instructions held in main memory.
- A machine instruction is a single instruction in a machine language program.
- The processor runs a program one machine instruction at a time, performing one operation at a time.
- All the machine instructions together make an executable program that does something useful.

Most programmers do not write programs in machine language. Most programs are created using high-level programming languages such as Visual Basic, C, C++ and Java. With a high-level language, the programmer creates a program in plain, human-readable text that contains very large and powerful operations. The operations will later be converted into many little machine operations.

For example, here is a line from a program in the Java language:

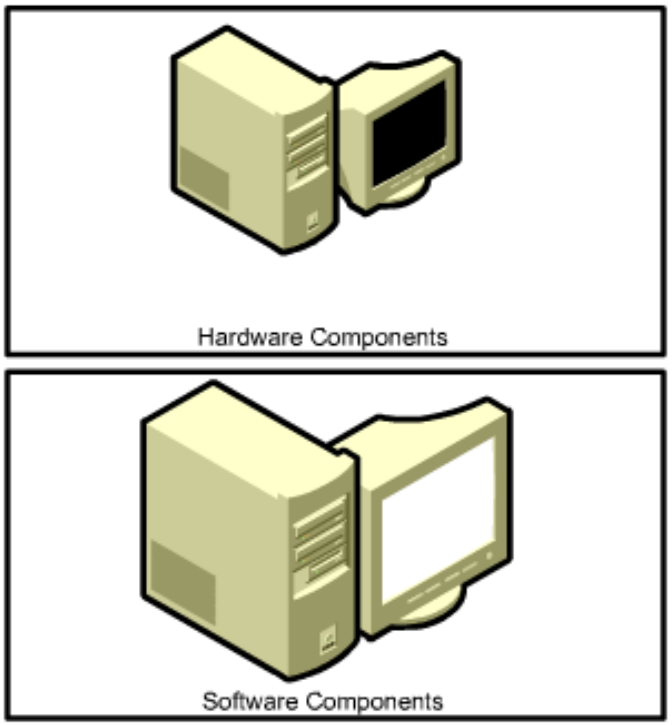
```
int sum = 10;
```

The machine operations for this line of Java code will:

- Set up a small part of main memory to hold a number.
- Store the number **10** in that section of memory.
- Assign a name to that **space** in memory so that other parts of the program can find and use the number that is stored there.

It might take a hundred or more machine operations to do all of this. Luckily, the Java programmer can ask for the processor to do all of these hundreds of operations just by writing the Java statement shown above.

Decide how each of the five images shown below should be categorized. *Then, click each item and drag it to the correct category indicated by the computers marked either Hardware Components or Software Components.* Don't forget to check your feedback to see how you did!



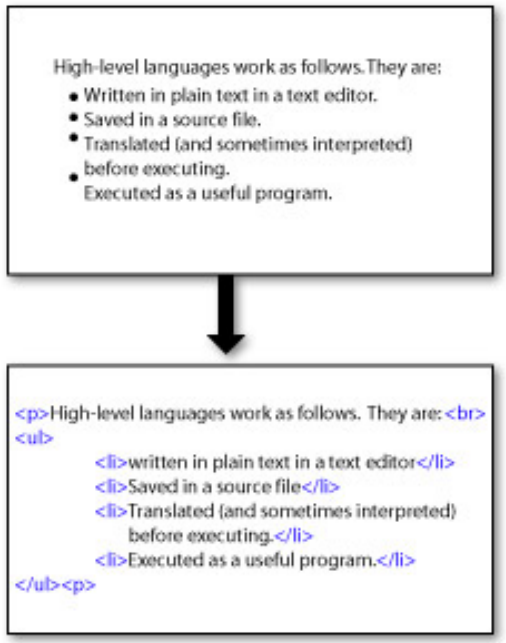
How High Level Programming Languages Work

High-level languages work as follows. They are:

- Written in plain text in a text editor
- Saved in a source file
- Translated (and sometimes interpreted) before executing
- Executed as a useful program

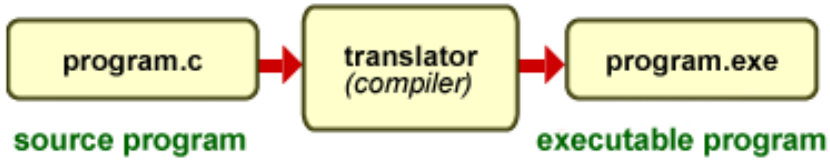
Source code:

- The programmer writes lines of text created with a text editor such as Notepad.
- The lines of text are saved in a file on the hard drive.
- These lines of text are called the source code.
- The file in which the source code is saved is the source file.
- The text in the source file can be edited with a text editor and printed on a printer.
- The source code does not contain any machine instructions so it cannot be executed (made to run) by the processor at this stage.



Translator:

- Source code is usually translated into a machine language program.
- An application program, called a **translator** or **compiler**, uses the source file as input and translates it into machine language instructions to produce an executable program as its output.
- At this stage, the executable version of the program can be copied into main memory and made to run.
- The word **compile** means the same thing as **translate**.



Interpreter :

- Some high-level languages, such as Java, add additional steps before the program can be executed.
- The extra steps involve an interpreter.
- The interpreter is located in the computer's main memory along with the program's compiled source code.
- There are different interpreters for different processors, such as Intel, Unix, Linux, Solaris, etc.
- The interpreter **acts like** the computer's processor, making it look like the computer's processor is executing the Java code.

You might wonder why this extra step involving an interpreter would be added. This extra step with the Java language adds a very desirable characteristic to the language that other popular languages, like C++, do not have. This characteristic is **portability**. This means that all Java programs can be run on any platform, unlike C++, as long as the Java interpreter for that platform is installed. When you compile a C++ program on your Intel Pentium computer, it will only run on the Intel platform. If you compile it on a UNIX platform, it will only run on the UNIX, etc. We will look at this term **portability** in the next lesson.

Portability

In an ideal situation, a program only has to be written once in a high-level language. The source file can then be translated into multiple executable files with each file containing the correct machine instructions for the processor on which it will run. For example, a source file is created and translated into an executable file for Pentium processors. Then, the same source file is translated into an executable file for Macintosh processors.

This idea of using one source file for executable programs that run on different processors is called **software portability**. Basically, software portability means the ability to write a program once in a high-level language and then being able to run it on any computer system regardless of the type of processor that system has.

Programs that are written in the Java programming language are portable. Programs written in C++ are not portable.

Memory

As we have already seen, the processor is the computer's **brain**, where all basic computing takes place. The processor has very little memory. It relies on the memory capacity of other components in the system to hold data and programs and to save results once data is processed.

There are two fundamental types of memory, **main memory** and **secondary memory**.

Characteristics of main memory:

- Closely connected to the processor
- Its contents can be quickly and easily changed
- Holds the programs and data that the processor is working with at the time
- Interacts with the processor millions of times per second

Characteristics of secondary memory (hard disks/removable storage):

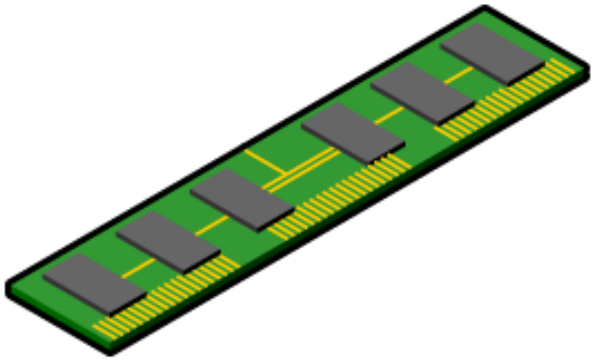
- Connected to main memory through the bus and a controller
- Contents can be easily changed but the change is slow compared to main memory
- Used for long-term storage of programs and data
- Only occasionally interacts with the processor

Main Memory

Main memory is where programs and data are kept when the processor is actively using them. Once a program and data becomes active, they are copied into main memory from secondary memory. Once in main memory, the processor can interact with them. A copy of the program and data still remains in secondary memory. Moving instructions and data from main memory into and out of the processor is very fast.

Nothing permanent is kept in main memory. Sometimes data is placed in main memory for a few seconds but the data stays there only for as long as it is needed.

Main memory is often referred to as **RAM** or **R**andom **A**ccess **M**emory. **Random** means that the memory cells can be accessed in any order. When you hear someone say that their computer has "128 megabytes of RAM," they are talking about how big their computer's main memory is.



Secondary Memory

Secondary memory is where programs and data are stored on a long-term basis. Hard disks and removable storage devices such as CDs and diskettes are the most common secondary memory storage devices.

Characteristics of hard disks:

- Large storage capacity, compared to main memory
- Usually found inside the computer system
- Used for long-term storage of programs and data
- Programs and data on the hard disk are organized into files.

Characteristics of removable storage:

- Used mostly for transferring software and data between computer systems or for back-up of software and data
- Very slow compared to other storage devices

The table below compares main memory and secondary memory:

Main Memory	Secondary Memory
Fast	Slow
Expensive	Cheap
Low storage capacity	Large storage capacity
Connects directly to processor	Not connected directly to processor

When a program is running, a section of main memory is allocated for the data the program is using. Memory allocations can change as many times as the program needs. For example, if a program is adding a list of numbers, the sum will be kept in main memory at a certain **location**. As new numbers are added to the sum, the sum will change and therefore, the memory location will have to be changed also.

Other sections of main memory might not change at all while a program is running. For example, the memory allocation for the instructions that make up a program does not usually change as the program runs.

When you write a computer program in most languages, you do not need to keep track of memory allocations or the contents of these locations. Part of the purpose of a programming language is to do these things automatically.

Secondary memory devices such as hard disks and floppy disks are used for long-term storage of information such as programs and data. Disk memory is usually organized into files. A file is a collection of information that has been given a name and stored in secondary memory.

Files can be quite large. Their size is measured in kilobytes or megabytes. One of the jobs of your computer's **operating system** is to keep track of file names and where they are located on the hard disk or floppy disk.

When a user calls for a program to execute, the operating system must find the program files that are needed and copy them into main memory. As the program runs, it asks for additional information stored in files on the hard disk. The operating system must locate these files and copy them into main memory.

When an application program is running, it is constantly asking the operating system to perform file manipulation tasks and then waits for these tasks to be completed.

We will study files in more detail in upcoming lessons.

Software

Software is the collection of programs and data that a computer uses. Software is stored on a hardware device such as a hard disk or CD-ROM.

Software consists of both programs and data. Programs are a list of instructions for the processor. Data include any information that the program needs. The data could be character data, numerical data, image data, audio data, etc.

There are two categories of software programs:

1. Application programs
2. Systems programs

Application Programs

Application programs, or applications, are the programs that people use to do their work. MS Word and MS Excel are examples of applications.

Systems Programs

Systems programs keep all the hardware and software running smoothly. The most important systems program is the operating system. Modern operating systems such as Windows come with a user interface. The user uses the interface to interact with application programs through windows, buttons, menus, icons, the mouse and keyboard. Other examples of operating systems besides Windows include Unix, Linux and Solaris.