MECH 215

Lecture 1

Computers and Programming Basics

Computers

- Computers are digital switching devices that use voltages to represent data, states and to control devices.
- In order to have the computer perform operations we provide the machine with a detailed set of instructions on what to do in a particular situation.
- These detailed instructions are typically called programs or code.

Computers

- The code that we write uses characters and symbols that represent particular things and operations.
- Typically the program or code will use a specific syntax (i.e. grammar that must be followed completely) to write each instruction.
- The syntax will identity how to represent variables and operations in the proper format so that the computer can interpret the instructions.

Computers

- The process of converting the program from a set of instructions (words written in a file) to operations that a computer can perform is called compiling.
- A compiler converts the code we write into a series of digital values (ones and zeros in binary number format) that the computer will use to perform the assigned tasks.
- Computers will follow absolutely the instructions as given to them, as such often it is very important to check what you have told them to do (de-bugging).

Computers

- The process of writing a program starts with outlining the steps that the program must take in order to produce the required results.
  - Pseudo code is one method of writing the steps using structured sentences to outline the steps.
  - Flow charts are graphical outlines used to represent program flow.
- Once the structure of the program is developed the writing of the program (coding) begins.

Programs

- Computer code is often written as lines of instructions in a text file.
- The text file must comply to any formatting requirements that the compiler has (e.g. spacing, forbidden characters, number of characters allowed per variable or per line).
- The text file is made using a text editor. There are many choices but you should be familiar with the ones available in the labs for this course, you will be using them.

Programs

- In the case of C++ there are some common elements for all programs.
- Compiler Directives: These are instructions that tell the complier to use certain libraries (these are collections of useful programs) when compiling the program.
- Main function: There is only one main function for each C++ program. The main program starts with the instruction int main().

Programs

- The line immediately after the int main () instruction is { . This indicates to the compiler the beginning of the program and the } character indicates to the compiler the end of the program.
- The instruction return 0 is placed just before the } character.
- The int main() refers to the value that the main will return at the end of the program this was used as a check.

Programs

- The return 0 instruction just before the end of the program returns (sends) a value of 0 to the operating system. Historically the zero was used to indicate success.
- Each instruction has a particular purpose and is composed of specific elements.
- Constants are used to represent fixed values, these may be numerical (X=5) logical (Danger=TRUE) or string (SCHOOL_NAME=CONCORDIA). Variables are used to represent values that may change.

Program

- Character: char (Individual characters or small integers).
- Integer: int, (Also short and long)
- Floating point: float, double, long double
- Related function sizeof

Programs

- In the C++ programming environment, both constants and variables are defined prior to use. These definitions are statements that specify the type of constant or variable, called assigning a value. For example a floating point value for pi would be written as:
  - float pi=3.14;
- Pi is assigned the floating point value 3.14. You can also define operations, operations are actions performed on constants or variables.

Programs

- Suppose I wanted to define the area of a circle using constants (for pi) and variables for the radius “r” and area “area” we could write the following code:
  - float pi=3.14;
  - float area, r;
  - r=0.500;
  - area=pi*r*r;
- In this case the first line defined the value for the floating point constant pi and set it to the value 3.14. The second line defined the variables area and r as floating point variables.

Program

- The next line sets the value of the variable r to 0.5. The last line defines the area as the product of pi times r squared.
- This assignation is a literal since the value does not change during the execution of the program.
- A variable (such as a counter) changes during the execution of the program.

Program

- Structure is a abstract data type that we will look at later on.
- Structure holds different data types pertaining to a single object or thing.
- An example is for your names we have an ID (integer value, a cumulative GPA (floating point), a yearly GPA (floating point) as well as an academic standing (string)). Each of these data types is computed and treated differently but are all assigned to you.

Program

- Structures can be nested and we will discuss that later on in the course.
- In addition to the language used, programming will require a methodology that will efficiently convert the problem into a series of steps that can be performed by the computer.

Programs

- Before starting a program it is a good idea to develop a structured approach.
- This is often called problem definition.
- In defining the problem and forming an approach to solving the problem, using a structured and well organized method can lead to more efficient solutions.
- There are many methods for problem definition and solution formulation, you should start developing one that works for you.

Programs

- Defining the problem: Write out a concise definition of what needs to be done in order to solve the problem.
- Structuring the solution: Write out steps that will provide the required elements of the solution.
- Start writing out the code for each step. Test that step and when it works move on to the next step.
- When you have written out the last step test the resulting code to verify that the program works.

Programs

- Commenting code
- It is a common practice to define all of the parts of the program with comments.
- Comments are words and phrases that the compiler will not look at but other people (including yourself) may look at with a text editor. These comments tend to be used to explain what a variable is and what a part of the program does. The // symbol is used to identify a line that the compiler should ignore.

Programs

- Alternately you can bracket comments by starting them with /* and ending them with */.
- There are many uses for comments, but the main one is to remind you what the program does and to allow others to make sense of what you are doing.
- This is a very good practice (often called documentation) that should employ as much as possible.

Next Lecture

- In the next lecture we will discuss in more detail assignment operations in C++