INTRODUCTION – COMPUTERS, BASIC C++, UNIX

ACKNOWLEDGEMENT: THE SLIDES ARE PREPARED FROM SLIDES PROVIDED BY NANCY M. AMATO AND JORY DENNY
Basic architecture
- Memory stores the data and instructions
- CPU is the brain
  - Arithmetic and logical operations
  - Controls
- Peripheral devices: I/O devices
Software layer: how program and user interacts with the computer
Data
- Type – defines possible actions (primitives, classes – user-defined, templates)
- Variable – instance in memory
- Declaration – Tell the compile variable, function, or type exists
- Definition – Initializing value of variable, full specification of type or function

Example
- `int i; //declaration : type of variable i is int`
- `Foo bar(10); //declaration + definition`
C++ REVIEW (CH 1.2)

- Operators:
  - Mathematical (+, -, *, /, %)
  - Comparison or Relational (==, !=, >, <, etc)
  - Assignment (=)
  - Access ([], (), *, ->)
  - Boolean or Logical (&, ^, !, &&, ||, etc)
C++ REVIEW (CH 1.3)

- Control Flow:
  - Branching:
    - If-else:
      ```cpp
      if( a <= b ) cout<<a;
      else cout<<b;
      ```
    - Switch:
      ```cpp
      switch(alphabet) {
        case 'A':
          cout<<"A";
          break;
        ...
      }
      ```
  - Loops:
    - Basic for
      ```cpp
      for(int i = 0; i < 10; ++i);
      ```
    - Basic while
      ```cpp
      while(!done) do_something();
      ```
    - Basic do while
      ```cpp
      do { something(); } while(!done);
      ```
Functions

Function Declaration and Basic signature: Let compiler know of the existence of the function

```
return_type function_name(param1, param2, ..., paramN);
```

Function Definition: Stating the function body

Example

```
void foo(int, char); // declaration
Foo bar(int i, char c); // declaration
void sum (int i, int j) { i = i+j; } // definition
```
- PuTTY is an application to establish SSH connections
- Development machine: linux.cse.tamu.edu
- Open a PuTTY session and log into linux
BASIC UNIX COMMANDS

- `cd`: change directory

- `mkdir`: make directory

- `ls`: list items in a directory

- `g++-4.7`: invoke GNU’s C++ compiler

- `vim / emacs / nano`: text editing
  - Quick start guide
  - Vim cheat sheet
  - Vim settings file – put in home directory labeled “.vimrc”
**LINUX TOOLS**

- **gdb** – debugging tool for Linux
- **valgrind** – memory leak detector/memory profiler
- **screen** – helpful when working remotely in a terminal. Saves the terminal session even if the network connection cuts out
- **LaTeX** – tool for creating documents
- **top** – monitor the system processes
EXERCISE

- Make a directory called `lab1` in your home folder
  - Type `mkdir lab1`
  - Type `cd lab1`
- Create a file called `hello.cpp` using `vim`
  - Type `vim hello.cpp`
  - Write `hello` world to the screen
    - Type `i` to go to insert mode then type the program
    - Type `esc` to go to command mode and `:wq` to save and quit `vim`
- Compile using `g++-4.7` and run your application
  - Type `g++-4.7 hello.cpp -o hello` to compile
  - Type `./hello` to run the program