1.1 Computer Systems

- **Hardware** -- Parts of a computer you can touch
  - PC
  - Workstation
  - Mainframe
  - Network
  - Input/output
  - Memory
    - Primary, secondary
    - Fixed, removable
  - CPU

- **Software** -- Parts of a computer you cannot touch
  - Operating Systems
    - Macintosh
    - Windows
    - Linux
  - High-Level Languages
    - Ada, C++, Java, BASIC, Lisp, Scheme
  - Compilers
    - Source program, object program, Linking
  - Editor: Integrated Development Environments (IDE)
    - IDEs combine editor, compiler and the execution environment (usually including a debugger)

1.2 Programming and Problem Solving

- **Algorithms**
  - Idea is more general than 'program'
  - Hard part of solving a problem is finding the algorithm
- **Program Design Process**
  - Problem Solving phase
  - Implementation phase
  - Algorithm Design
  - Desk Testing
  - Testing
  - Working Program

Software Life Cycle

- Analysis and specification of task (problem definition).
- Design of the software (algorithm design)
- Implementation (coding).
- Testing
- Maintenance and evolution of the system
- Obsolescence

1.3 Introduction to C++

- **Origins of the C++ Language**
  - Bjarne Stroustrup designed C++ for modeling.
  - C++ is an OOP extension of the C language.
  - C was developed as a systems programming language from the B language in the Unix environment. It grew into a general purpose programming language as its libraries were developed.
### Display 1.8 A Sample C++ Program

```cpp
#include <iostream>
using namespace std;

int main()
{
    int number_of_pods, peas_per_pod, total_peas;
    cout << "Press return after entering a number:\n";
    cin >> number_of_pods;
    cout << "Enter the number of peas in a pod:\n";
    cin >> peas_per_pod;
    total_peas = number_of_pods * peas_per_pod;
    cout << "If you have \n";
    cout << number_of_pods;
    cout << " pea pods\n";
    cout << peas_per_pod;
    cout << " peas in each pod, then\n";
    cout << total_peas;
    cout << " peas in all the pods.\n";
    return 0;
}
```

### Some Details of a C++ Program

```cpp
#include <iostream>
using namespace std;

int main()
{
    // These lines are a complicated way to say "the program starts here"
    return 0;
}
```

### Some Details of a C++ Program

- `cout` is the output stream, read See-Out. It is attached to the monitor screen. `<<` is the insertion operator.
- `cin` is the input stream, read see-in, and is attached to the keyboard. `>>` is the extraction operator.
- "Press return after entering a number:\n" is called a cstring literal. It is a message for the user.
- `cout << "Press return ...\n"` sends the message to cout
- `cin >> number_of_pods;`

### Some Details of a C++ Program

- `cout << "Enter the number of peas in a pod:\n";` sends a request to the user.
- `cin >> peas_per_pod;`
- The second extracts an integer value (the type of `peas_per_pod`) into `peas_per_pod`.

### Some Details of a C++ Program

- `total_peas = number_of_pods * peas_per_pod;`
- The asterisk `*` is used for multiplication.
- This line multiplies the already entered values of `number_of_pods` and `peas_per_pod` to give a value which is stored (assigned to) `total_peas`. 
#include <iostream>
using namespace std;

int main()
{
    variable_declarations
    Statement1;
    Statement2;
    . . .
    Statement_last;
    return 0;
}

An include directive
More later, for now "do it"
Declares main function
Start of main’s block
Says "end program here"
End of main’s block

You write a C++ program using a text editor exactly as you
would write a letter home.
Compiling depends on the environment.
You may be using an Integrated Development Environment
or IDE. Each IDE has its own way to do things.
Read you manuals and consult a local expert.
You may be using a command line compiler. In that event,
you may some thing like write:
cc myProgram.cpp
Your compiler may require .cxx .cc or perhaps .C
Linking is usually automatic. Again, read your manuals and
ask a local expert.

Design errors occur when specifications
are do not match the problem being solved.
The compiler detects errors in syntax
Run-time errors result from incorrect
assumptions, incomplete understanding of
the programming language, or
unanticipated user errors.

Hardware physical computing machines.
Software programs that run on hardware.
Five computer components: input and output devices,
CPU, main memory, secondary memory.
There are two kinds of memory: main and secondary.
Main memory is used when program is running.
Secondary memory is usually nonvolatile.
Main memory is divided into bytes, usually individually
addressable.
Byte: 8 bits. A bit is 0 or 1.