

# Lab 1

---

Your name:

Your email address:

Your student ID number:

---

## Lab Objectives

To gain experience with

- the activity of programming
  - the architecture of computers
  - using your compiler
  - recognizing syntax and logic errors
  - writing algorithms in plain English
- 

## R1. Becoming familiar with your computer

All modern computers offer their users an interface to their physical, electrical, and digital systems. Your computer's *operating system* provides fundamental services from behind the scenes. A large part of a computer's work is simply storing and retrieving *data* so that it can be available to use.

Put simply, the operating system takes care of a lot of the filing work - storing, moving and remembering where things are - so you can do useful things at a higher level. You interact with the operating system via a *Graphic User Interface (GUI)*, for example a windowing system using a pointing device like a mouse, or a *Command Line Interface*, using only a keyboard and a single window.

What are some examples of tasks you'd tell the computer to do via the Operating System interface?

Run an application, close applications, change settings, and search for files.

Operations like browsing the web and editing text in a word processor - most of the things you'd have the computer do - involve your operating system executing some *program*. The Operating System

itself is a program which is running all the time. It executes your instructions and, in turn, it can run other programs.

It is useful to think of a program as a sequence of instructions. Both executable instructions and digital data can be represented as files, for example, on your hard drive. Both are sequences of symbols, just like the letters that make up the words in this sentence. Your job as a programmer is of course to provide the instructions that operate on the data.

We will begin to get some experience with the Operating System by finding a file. We will locate `math.h`, an include file used by C++. You may need to ask your instructor or a friend how to find a file on your computer as well as the location of the include files.

What did you do to find `math.h`?

```
I used the search bar built in to Mac OS X.
```

How did you open `math.h` ?

```
I clicked on the link and it was opened in Xcode window (I use Xcode for C++  
programing).
```

The program that you'll be running to write computer programs is your text editor, which often is part of an integrated compiler environment. You may be working in a computer lab--then ask your lab guide how to start the editor. Or you may have purchased and installed your own compiler. Then follow the vendors instructions. Go ahead and start it now.

Again, what did you do?

```
I opened the Xcode program that was built in to Mac OS X.
```

---

## R2. Compiling and running programs from within your development environment

Frequently in these labs you will be asked to compile a sample program. Below is a copy of a C++ program that displays a drawing. Copy and paste it into your compiler's editor, and from there save it as `art.cpp`.

Describe what you did.

I highlighted the text under this box, hit CMD+C, then CMD+Tab to get to Xcode, then CMD+V to paste the text.

```
#include <iostream>
#include <string>
using namespace std;
int main()
/* PURPOSE: Display an 'art' drawing
*/
{
    string s1 = " *   *   *   *   *   * ";
    string s2 = "  *   *   *   *   *   ";
    string s3 = "_____ \n";
    string s4 = "_____ \n";

    cout << s4 << s1 << s3 << s2 << s3;
    cout << s1 << s3 << s2 << s3;
    cout << s1 << s3 << s2 << s3;
    cout << s1 << s3 << s2 << s3;
    cout << s4 << s4 << s4 << s4 << s4;

    return 0;
}
```

Once you have typed in (or, in this case, pasted in) a program, you need to to *compile* it to to create an executable file. Again, these steps depend on your compilation environment. Find out the steps for your computer system, then go ahead and compile `art.cpp` to an executable file.

Describe what you did.

I hit CMD+R to build and run the program.

Finally, *execute* the program. Once again, the steps depend on your computer system.

Describe what you did to execute the program.

I hit CMD+R to build and run the program.

Describe what happened when the program executed.

An text image (I would say ASCII image, but Xcode uses Unicode) of the American flag appeared in the standard output.

---

## P1. Writing simple programs

Your initial C++ programs will be contained entirely in one file, and there are some elements that they all will have because of requirements of the C++ language. Essentially, these are 1) markers for the compiler - to know where your program begins and ends - and 2) files that your program also needs in order to operate. When you build a program, your compiler looks for code of the form:

```
#include <iostream>
using namespace std;
int main()
{
    /*
       your work goes here
    */

    return 0;
}
```

The textbook has a program that prints the message `Hello, world!` on the screen.

Try changing it to display `Hello, Universe!`

Type the program into your compiler's editor, compile and test. Then paste the source code in the following text box.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    cout << "Hello, Universe!";

    return 0;
}
```

---

## P2. Detecting syntax and logic errors

There are numerous opportunities for error in any program, many times in places that seem too simple to require close attention. What do you think the following program is supposed to do?

The program is supposed to calculate and print the volume and area of a sphere of given radius.

```
#include iostream.h

int main()
{ double radius = 11; /* centimeters
  double pi  = 3.14
  double sphere_volume = (4/3)*pi*(radius * radius ** radius);
  double surface_area = 5 * pi * radius;
  cout << "Volume = " < sphere_volume;
  cout << "Area = " << surface_area;
  return 0;
}
```

Will it work as it is?

No, because a multi-line comment is started but never ended after the declaration of "radius". Even if that comment weren't there, there is a missing

Try compiling the program. What were the results? (Use copy and paste to place a copy of your compiler's error messages here.)

Received compiler errors:

Expected "FILENAME" or <FILENAME>

Fix the syntax errors. Place a copy of your program's output here.

```
Volume = 4179.34Area = 172.7
```

The program has two logic errors. Fix them both and paste the corrected program here.

```
The formula for the surface area of a sphere is 4(pi)(radius^2), not  
5(pi)(radius).
```

---

### R3. Algorithms

An algorithm is a recipe to carry out a task. To be programmable, such a recipe must be

- Unambiguous
- Executable
- Terminating

While an algorithm will eventually be expressed in a particular programming language, it is frequently helpful to describe its operation in natural language, before undertaking the task of coding it up. Suppose for example that you would like to calculate your future salary, knowing that you will receive a 3% raise every year. If you made \$20,000 this year, your salary next year would be \$20,600 ( $\$20,000 + (\$20,000) \cdot .03$ ) and \$21,218 the year after ( $\$20,600 + (\$20,600 \cdot .03)$ ), \$21,854.54 the year after ( $\$21,218 + (\$21,218 \cdot .03)$ ), etc.

Describe an algorithm to do this computation for a salary of \$30000 after any number of years. Your algorithm should be so detailed that anyone can carry out the steps and arrive at the correct answer.

```
For each year, take the previous year's salary and add to it that same salary  
time .03.
```

---

### P3. Additional Programming Practice

Often times, you will find that a program can be written by modifying another program that you have written before. The calculation which we just wrote the algorithm for can be generalized as follows:

$$P = C (1 + r)^t$$

Where P is the final pay rate, C is the starting pay rate, r is the percent increase (expressed as a decimal) and t is the number of years. So, for example, to compute your salary after 5 years, starting at \$20000 with 3% increases per year we would get:

$$P = 20000(1+.03)^5 = 23185.48$$

Modify the program that you fixed in section P2 to calculate your pay after being with a company for 10 years. Assume that you had started with a salary of \$18,000 and that you receive a 3% raise every year. having started with a salary of \$18,000. Paste your program here:

```
#include <iostream>
#include <math.h>

using namespace std;

int main()
{   double startingSalary = 18000;
    double rate = .03;
    double time = 10;
    double finalSalary = startingSalary * pow((1 + rate),(time));
    cout << "Final Salary = " << finalSalary << "\n";
    return 0;
}
```

Now, compile it and execute it. Paste your output in the space below:

```
Final Salary = 24190.5
```

Don't forget to save this page and then up load it to moodle when you're finished.