

Background

Most programming involves reading, writing, and manipulating data. Later in the semester we will talk about ways to create complex data structures for storing information, but for now we will focus on the basic data types which C++ provides by default. The table below shows some of the most important built-in types and explains what they are for:

Type	Use
int	Integer Values
char	Single Characters
double	Floating point values (decimals)
bool	True or False values
string	Strings of Characters

To store information, you create a variable of a particular type. For instance, to create an integer value named "x" you can type:

```
int x;
```

To create a string named "address" you can type:

```
string address;
```

Once you have declared a variable, you can use it to store information. For instance,

```
x = 10;
```

will set x to the value 10. The assignment

```
address = "911 West Fire Station Street";
```

will store information in address. Notice the double quotes in the statement above. These are necessary so that C++ knows that the text is a string constant and not a variable name or keyword.

We can use variables as a temporary place to store the result of a computation. The code

```
x = 5 + 7;
```

will add the numbers 5 and 7 and store the result in the variable named "x". You can also ask C++ to use the value that is currently stored in the variable to perform calculations. The line

```
x = x * 9;
```

will retrieve a value from "x" (currently 12), multiply it by 9, and then store the result back into "x". The variable will now hold the value 108.

Preparation

Create a directory named "Project1". In that directory, make a file named "project1.cc". In that file type the following code:

```
#include <iostream>
#include <iomanip>

using namespace std;

int main(int argc, char** argv){

}
```

This code pulls in some libraries that other people have written (in this case, the input and output functions of the standard C and C++ libraries). It then tells the compiler to use the standard namespace and creates a function called "main".

The main function is where our code is going to go. Later, we will learn how to create and use our own functions. But for now, all of the code you type is going to go between the curly braces that delineate the main function.

Creating an Integer

At the top of the main function, on the line below the first curly brace, create an integer variable named "result" (review the introduction if you don't remember how to do this). Then use the assignment operator (i.e. the = operator) to set the value of result to 100.

Now create a floating point value named "pi" and a string variable named "nickname". Set pi to 3.14159265 and nickname to "bozo".

Printing to the screen

Having variables doesn't do us much good if we can't display the information that is stored in them. C++ handles input (reading) and output (writing) using *streams*. A stream is a special kind of variable that can send or receive information using the input and output operators. There are three default streams that C++ provides for you to use. The "cin" stream allows you to read data from the keyboard one line at a time. The "cout" stream allows you to print data to the screen. The "cerr" stream allows you to print error messages.

At the bottom of your main function (but before the closing curly brace), add the following line:

```
cout << "Hello, World!" << endl;
```

Save your file and quit. You have written your first C++ program!

To compile it, type "g++ project1.cc -o Project1" and hit enter.

To run your program type "./Project1" and hit enter.

Outputting Variables

The "Hello, World!" text we printed was a simple string constant. We can also tell C++ to print the values of our variables. Below the hello world line, add the following lines:

```
cout << "Result: " << result;
cout << " Pi: " << pi;
cout << " Nickname: " << nickname << endl;
```

Compile your program again and run it. Notice that all three variables show up on one line! C++ doesn't wrap lines unless we send the end-of-line character to the output stream. That is what the "endl" is for. It is a special kind of string constant that tells C++ to start a new line.

Reading from the keyboard

To read from the keyboard, we use "cin". Add the following lines to the bottom of the main function:

```
cout << "Enter a new nickname: " << endl;
cin >> nickname;
cout << "Your new nickname is " << nickname << endl;
```

Compile and run your program again. It should ask you for a new nickname and then print out what you typed in.

Notice that the arrows for the cin face the opposite direction from those for cout. The arrows point in the direction of information transfer. With cout, we are sending the string TO the screen so we use "<<". With cin we are reading the information FROM the keyboard so we use ">>".

Submitting your work

In order to submit your work, you must first create a tarball of the entire Project1 folder. To do this, you must back out of the folder and access the parent folder. The command for this is "cd .."

Then you should type exactly the following:

```
tar czpvf Project1.tar.gz Project1/
```

This tells the computer to create a "Tape ARchive" file and compress it. The name of the file is "Project1.tar.gz" and it will contain everything in the Project1 folder. Now open up a web browser to <http://narnia.homeunix.com/~robert/submit/> select CS160 in the list and type in your username and password. You should be able to submit the Project1.tar.gz tarball in the same way you uploaded Project 0.