What a program typically looks like

CS 10A – INTRO TO C++

Basic Layout

```
#include <iostream>          // Always include this line
using namespace std;         // Always include this line
int main()   {               // Begin main part of the program


                             // This is where your code goes
                             // Runs top to bottom, line by line


        return 0;            // End of the program
}
```

Outputting Text

Program

```
int main()
{
        cout << “Hello World!”;

        // cout = console output
        // Text is always in quotes

        return 0;
}
```

Console

- ./a.exe

Hello World!

Outputting Text

Program

```
int main()
{
        cout << “Hello World!” << endl;
        cout << “Hello again!”;

        // endl = end line, goes to next one
        // can also be written in one line

        return 0;
}
```

Console

```
➢ ./a.exe
Hello World!
Hello again!
```

Programming Tips

- Comments
  - You can write notes that are unseen by the compiler in your code. Highly useful and should be used often for reference.
  - // allows you to comment out one line
  - /* comment */ allows you to comment out entire blocks of text
  - On these slides, yellow comments will indicate more crucial notes.
- White Space
  - C and C++ both ignore how you use spaces, tabs, and new lines in your code. Use this fact to write easy to read code.
  - What matters is how you use your other symbols: quotes, parentheses, brackets, semicolons, etc.

Comments

```
int main()
{
        // This is a one line comment. The compiler never sees what's behind these slashes.

        /*
           This is a comment block. This allows you to write comments that take up multiple lines.
           Anything that exists between the asterisks will not be seen by the compiler.
        */

        /*
        * This is another way to make comment blocks for easy reading. Some IDEs may do this
        * automatically. Like before, any text between the first and last asterisks is a comment.
        */
        return 0;
}
```

White Spacing

This program is the same…

```
int main()
{
        cout << “Hello World!” << endl;
        return 0;
}
```

…as this program.

```
int main()
{
        cout <<
                “Hello World!”
                <<
                endl;
        return 0;
}
```

Escape Characters

- \t – tab
- \n – new line, identical to endl
- \” – allows you to use a double quote in your actual text
- \’ – allows you to use a single quote in your actual text
- \b – backspace
- \r – carriage return, moves cursor back to start of line
- \a – alert, a sound will play (on Windows, it’s that ding)
- …and more that you’ll probably never use in your lifetime

How to Use Escape Characters

Program

```
int main()
{
	cout << “Hello World!\n”;
	cout << “Hello again!\n:D”;
	return 0;
}
```

Console

```
➢ ./a.exe
Hello World!
Hello again!
:D
```

Variables

- Just like in math, variables are storage units for a certain value in programming. There are different types.
- What we'll start with: int and char
  - int = integer, stores a whole number
  - char = character, stores a single character
- Variables must declared with a name before they can be used. Any name is legal within certain restrictions:
  - Variable names must begin with a letter or an underscore
  - Spaces and special characters are never allowed
  - Keywords reserved by C++ are also banned
  - Note that variable names are case sensitive
  - Names must be less than 255 characters long, (recommended < 31)

Declaring and Assigning Values to Variables

```
int i = 0;                      // initialize a variable with a value
char ch0, ch1, ch2;             // declare multiple variables
int main()
{
        int x = 0, y = 1;       // declare multiple variables w/ initial values
        ch0 = 'a';              // assign a value to an existing variable
        ch1 = ch2 = 'b';        // assign the same value to multiple variables

        //  Variables can be declared inside or outside the main block

        return 0;
}
```

Using Variables

Program

```
int i = 1;
int main()
{
        cout << i << endl;
        cout << i*5 << endl;
/*
        Variables can only be used
        after declaring them.
*/
        return 0;
}
```

Console

➢ ./a.exe

```
1
5
```

Getting Inputs

Program

```
int main()
{
        int x;
        cout << “Enter a number: ”;
        cin >> x;
        cout << “You entered in ”
                << x << endl;
        // cin = console in
        // User presses enter to move on
        return 0;
}
```

Console

- ./a.exe

```
Enter a number: 26
You entered in 26
```