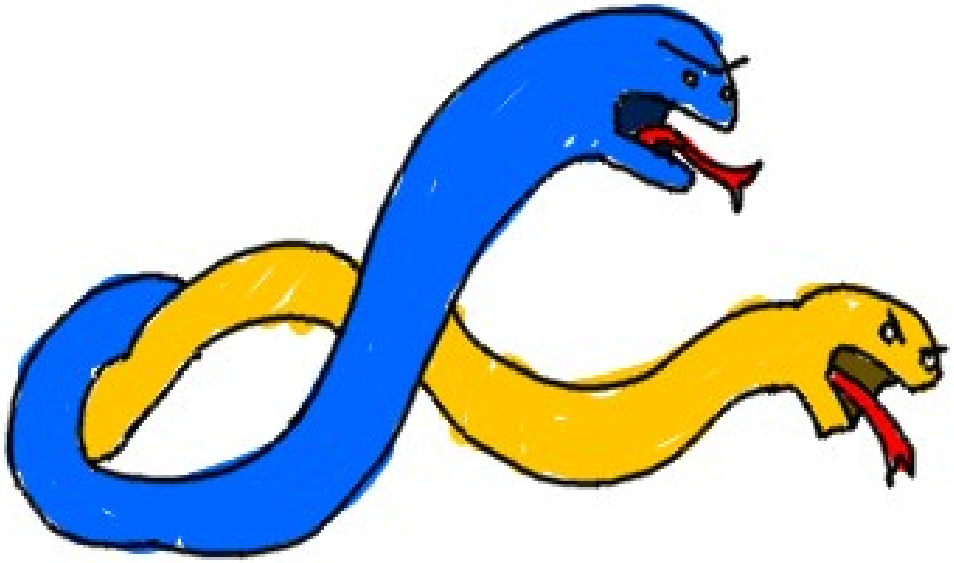
Python vs. C++



Written by Rachel J. Morris, last updated 2013-05-24

File I/O

- File Input and Output in C++ is similar to Input and Output to the Console, in that both are handled with streams.

```
// Console Output
cout << "Hello" << endl;

// Input from Console
cin >> username;
```

```
// File Output
outputFile << "Hello" << endl;

// Input from File
inputFile >> username;
```

File I/O

- When you're reading from or writing to a file, you must:
  - **`#include <fstream>`** library
  - Create a **ifstream** object (for input), or an **ofstream** object (for output)
  - Open a text file
  - Write or read the data
  - Close the text file when done

Writing to a file

```
#include <fstream>
using namespace std;

int main()
{
    ofstream outFile;
    outFile.open( "simple-out.txt" );

    outFile << "This is a simple file" << endl;
    outFile << "This has the same functionality as" << endl;
    outFile << "if you were using cout" << endl;

    outFile.close();

    return 0;
}
```

Reading from a file

```
#include <fstream>
using namespace std;

int main()
{
    ifstream inFile;
    inFile.open( "data.txt" );

    string name;
    int age;
    float balance;

    infile >> name >> age >> balance;

    inFile.close();

    return 0;
}
```

Reading from a file

- Like with **cin**, C++ uses **whitespace** as a delimiter. When reading from a file, it will automatically read one word (or number, ie 2, 24, 100) at a time.

Reading an entire file

- You can use a **while** loop to continue reading a file until the end.
- If the conditional statement
  `( inFile >> ages[index] )`
  returns false, that means there are no words left in the file.

```
int ages[100];
int index = 0;

while ( inFile >> ages[ index ] )
{
   index++;
}
```

More than just .txt

- Many files are just text files that follow a specific style. When you output to a file, you can create more than just a plaintext (.txt) file. You can create...
  - A source-code file: .cpp, .py
  - An HTML file: .html
  - A CSV file: .csv (a basic spreadsheet format)
  - An XML file: .xml
  - And more!

File I/O

Additional Reading

- Chapter 2.4 of the Pearson textbook
- Python vs. C++ class resource
  https://github.com/Moosader/Python-vs-CPP
- http://www.cplusplus.com/doc/tutorial/files/
- http://www.learncpp.com/cpp-tutorial/136-basic-file-io/

Random Numbers

- Random Numbers are another handy thing to know early on. In Python, we generate random numbers with:

```
import random

# Generate a random number between 1 and 10
randNumber = random.randint( 1, 10 )
print( randNumber )
```

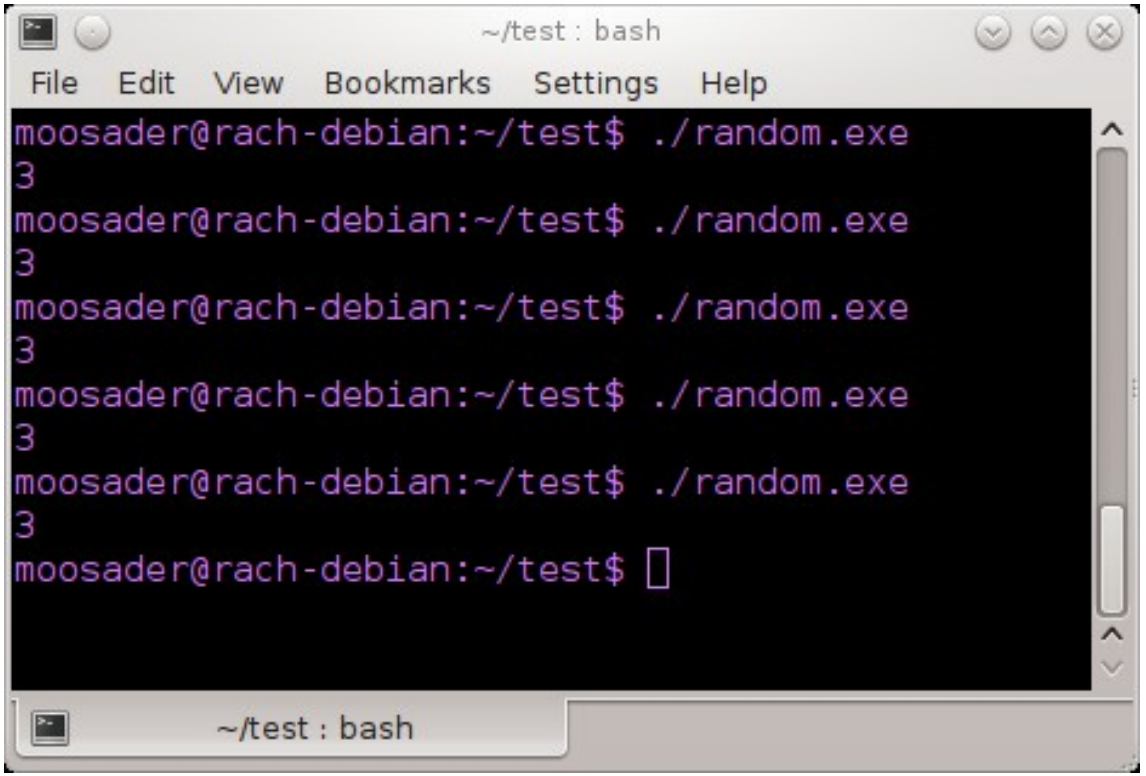
Random Numbers

- In C++, we will need to **#include** the **stdlib.h** library.

```
#include <stdlib.h>
#include <iostream>
using namespace std;

int main()
{
    // Generate a random number
    // between 0 and 9
    int randNum = rand() % 10;

    cout << randNum << endl;

    return 0;
}
```

Random Numbers

- However, if you run this program multiple times you will see that you keep getting the same number over and over!



Random Numbers

- Random Numbers generated by a computer aren't ***truly*** random.
- To make our numbers appear more random, we need to **seed** the random number generator.
- Often, the current time is used as the random seed, which ensures that each time the program is run it has a new seed.

Random Numbers

```
#include <stdlib.h>    // for random
#include <time.h>  // for time
#include <iostream>
using namespace std;

int main()
{
    // Seed the random number generator
    srand( time( NULL ) );

    // Generate # between 0 and 9
    int randNum = rand() % 10;

    cout << randNum << endl;

    return 0;
}
```