Discussion Session
Week 4

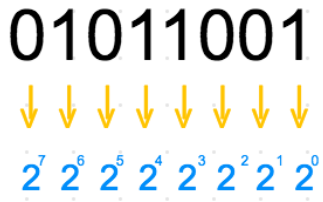
Exam #1 Review - Basics of Programming, Logic, and C++

General Data Information

- Data is stored in bits and bytes
  - 1 bit is the smallest unit of data (0 or 1)
  - 4 bits = 1 nibble
  - 8 bits = 1 byte
  - 1024 bytes = 1 kilobyte
  - 1024 kilobytes = 1 megabyte

Representation of Numbers

- Decimal (base 10)
  - Numbers you're familiar with
- Binary (base 2)
  - Powers of 2 and add
  - Can be x bits long, powers increase from right to left
  - 01011001 = 1 + 0 + 0 + 8 + 16 + 0 + 64 + 0 = 89
- Hexadecimal (base 16)
  - Powers of 16 and add
  - 0-9, A-F
  - 7B = 11 + 112 = 123
- Octal (base 8)
  - Powers of 8 and add



Base Conversions (From Decimal)

- To Binary
  - Divide the given number by 2, take the remainder, repeat
  - Write remainders backwards
  - 14
    - 14/2 = 7 r0, 7/2 = 3 r1, 3/2 = 1 r1, 1/2 = 0 r1
      - Binary 14 = 1110
- To Hex
  - Same process, dividing by 16 instead
- To Octal
  - Same process again, dividing by 8
- Simplest way from one base to another is to go through base 10

Background on C++

- C++ is a compiled language
  - There are many compilers, g++ is a common one
  - Code is translated into machine language for you
  - Any syntax errors will prevent successful compilation
- C++ is essentially C with libraries
  - Object oriented capabilities
  - Manual memory management
    - No garbage collection
- Everything in C++ can be boiled down to bits of information, and everything is treated as either true or false.

Basic C++ Programming

```
1  int main(int argc, char** argv) {
2    return 0;
3  }
```

- All C++ programs require a main function in order to run
- Main (usually) returns an integer and (usually) takes in two parameters, argc and argv
- In general, the value returned from main indicates the error status of a program (0 means successful exit by standard, non 0 denotes unsuccessful)
- C++ is strictly typed
  - Types of variables and return types of functions must be stated explicitly unlike Python
- Lines of code are ended with semicolons
- Comments can be written like so:
  - //Single line comments
  - /*

    Multi Line comments

    */

C++ Data Types

- Numbers
  - int
    - Signed integer values, 32 bits
    - 1st bit is the “sign” bit → 2^31 - 1 is the maximum value
    - Modifiers
      - Long, short, unsigned
        - Change the max/min value, number of bits stored in an int
        - Unsigned long long is 64 bits
  - float/double
    - Decimal values (varying precision)

C++ Data Types (cont)

- char
  - Characters (‘h’, ‘e’, ‘l’, ‘l’, ‘o’, etc)
- bool
  - True/False
- void
  - Valueless
    - Used as return types for functions that do not return values, or for polymorphism

Variables and Functions

- Variables are a means of storing data
  - Syntax
    - dataType variableName = value;
- Functions are blocks of code that can be repeated by calling them
  - Syntax
    - functionReturnType functionName(paramOneType paramOne, paramTwoType paramTwo...)

      {

      //Function body

      }

Logic (Truth Tables)

- Logic is the basis of programming

X	Y	AND(X,Y)	OR(X,Y)	NAND(X,Y)	NOR(X,Y)	XOR(X,Y)
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0

Logical Operators (C++)

- &&
  - And
- ||
  - Or
- !
  - not

Other Operators

- +, -, *, /
  - Addition, subtraction, multiplication, division
- %
  - Modulo
- <, <=, >, >=
  - Less than, less than or equal to, greater than, greater than or equal to

Conditionals

- If else if else statements
  - The classic conditional branch
- Switch Statements
  - Used for many different cases of a condition
  - Must have “default” case and “breaks”

```
int number = 5;

if (number > 0) {
  // code
}
else {
  // code
}

// code after if...else
```

```
9   int switchval = 0;
10
11  switch (switchval)
12  {
13      case 0:
14          cout<<"code path executed for value 0";
15          break;
16      case 2:
17          cout<<"code path executed for value 2";
18      case 3:
19          cout<<"code path executed for value 3";
20          break;
21      default:
22          cout<<"code path executed for default path";
23  }
```

Loops

- While loops
- For Loops
  - Classic, Range Based
- Do While Loops

```
while(condition)
{
  //Execute code while condition is true
}
```

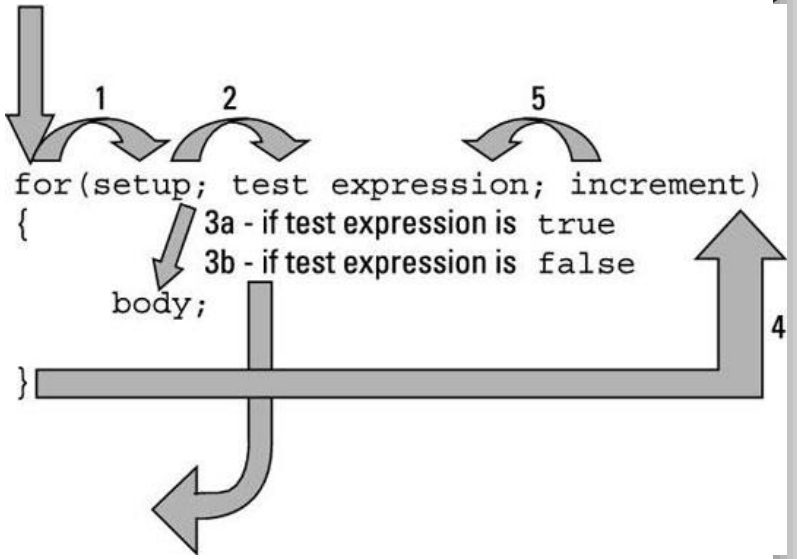
```
for(int i = 0; i < 10; ++i)
{
  //Execute code while i < 10
}
```

```
for(int i: integerArray)
{
  //iterate through an array of integers, i will be each value
}
```

```
do
{
  //Execute this block at least once, repeat while condition is true
} while (condition);
```

For Loops Expansion

- More can be done with for loops
- “setup” , “test expression”, “increment” can have really any code there, but it is always executed in the given order



The “++” Part

- Libraries can be included into your files using #include
  - #include <libraryName>
  - #include “filename”

Tracing Code

- When tracing code, we go sequentially and change data as old to
- http://pythontutor.com/visualize.html#mode=edit
  - PythonTutor is a great resource for practice in tracing code
  - Write some programs like we've done in the assignments and view the stack while it executes for some great exam prep