

CISC 1110 (CIS 1.5) Introduction to Programming Using C++

Spring 2012
Instructor: J. Raphael

Email Address: raphael@sci.brooklyn.cuny.edu
Course Page: <http://www.sci.brooklyn.cuny.edu/~raphael/cisc1110.html>
Class Hours: TTH 2:40 – 4:45PM 5122N

CISC1110-Spring2012-Raphael-lec4

Agenda

- Lab 2 & Programming Tips
- cin statement
- Control Structures
- Relational Operators & Logical Operators
- if statements
- Using brace { ... }
- Random numbers

CISC1110-Spring2012-Raphael-lec4

Lab 2

Good:

- Formatting (spacing and indentation)
- Variable names
- Output with messages
- Team work

Bad:

- Not saving work (where are the flash drives)
- Not saving/compiling often enough

CISC1110-Spring2012-Raphael-lec4

Programming Tips

- Save & Compile often
- Flash drives
- Basic Program Structure
- Experiment with language, **read errors**
- Comments

```
/*
   Name
   Assignment
   Class
*/
```

CISC1110-Spring2012-Raphael-lec4

Reading Data from the Keyboard

- `cin` is an object of class `istream` that represents the standard input stream.
- By default, most systems have their standard input set to the keyboard
- We write characters to it using the insertion operator (`ostream::operator >>`)

```
cin >> ... >> ... >> ... ;
```

Examples

```
cin >> age;
```

```
cin >> price >> grade;
```

CISC1110-Spring2012-Raphael-lec4

Sample Code

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

int main()
{
    int age, month;
    float payRate;
    char grade;

    // Entering data using multiple cin statements
    cout << "Enter pay rate ";
    cin >> payRate;

    cout << "Enter age and month of bday ";
    cin >> age >> month;
    cout << endl << endl << "You are " << age << " y.o. and born on " << month << endl << endl;

    // cin statement with multiple parameters
    cout << "Enter all 3 in one line (Age,Month,Pay,grade): ";
    cin >> age >> month >> payRate >> grade;

    cout << endl << endl << "Age " << age << endl;
    cout << "Month " << month << endl;
    cout << "Pay Rate " << payRate << endl;
    cout << "Grade " << grade << endl << endl;

    return 0;
}
```

CISC1110-Spring2012-Raphael-lec4

Control Structures

- Allows programs to bifurcate, repeat code or make decisions
- Compound-statement, sometimes called a block, is a group of statements separated by ';' and bounded by braces '{}'
- `if-else...`, `switch`, `goto`, loops are some of the control structures in C++
- `if` statements allows a program to make decisions
- A block of code is executed if a condition(s) is met
- There are 3 forms of `if` statements
 - `if`
 - `if-else`,
 - `if-else-if`

CISC1110-Spring2012-Raphael-lec4

Relational Operators

- Relational operators are used to compare two values
- They can be used to compare numbers or characters
- Comparing characters uses the ASCII Codes
- The relational operators look like operators in math, except for equality:

<code>==</code>	equality	<code>!=</code>	inequality
<code>></code>	greater than	<code><</code>	less than
<code>>=</code>	greater than or equal to	<code><=</code>	less than or equal to

CISC1110-Spring2012-Raphael-lec4

Logical Operators (!, &&, ||)

- ! NOT
- && AND
- || OR

p	q	p && q
T	F	F
T	T	T
F	F	F
F	T	F

p	q	p q
T	F	T
T	T	T
F	F	F
F	T	T

p	!p
T	F
F	T

CISC1110-Spring2012-Raphael-lec4

if Statement

```
if (condition) {
    // execute this block
}
```

• example

```
if ( x > y ) {
    cout << "x is bigger than y\n";
}
```

CISC1110-Spring2012-Raphael-lec4

if-else Statement

```
if (condition) {
    // execute this block when the condition is true
}
else {
    // execute this block when the condition is false
}
```

• example

```
if ( x > y ) {
    cout << "x is larger" << endl;
}
else {
    cout << "y is larger (or the same as x)" << endl;
}
```

CISC1110-Spring2012-Raphael-lec4

if-else-if Statement

```
if (condition1) {
    // executed when condition1 is true
}
else if (condition2) {
    // executed when condition2 is true
}
else if (condition3) {
    // executed when condition3 is true
}
.
.
.
else {
    // executed when all the above are false
    // default statement
}
```

CISC1110-Spring2012-Raphael-lec4

To Brace or Not To Brace...

• Example #1

```
int speed = 30;
if (speed < 55)
    speed = speed + 30;
    speed = speed / 2;
cout << endl << "Your speed is " << speed << endl;
```

• Example #2

```
int speed = 80;
if (speed < 55)
    speed = speed + 30;
    speed = speed / 2;
cout << endl << "Your speed is " << speed << endl;
```

CISC1110-Spring2012-Raphael-lec4

Random Numbers

- `int rand (void);`
- `void srand (unsigned int seed);`
- Not really random!!

i.e.

```
randomNumber = rand() % 100 + 1;
```

- Generates a number between 1 and 100, inclusive
- In general if $r = \text{rand}() \% n + t$ then $(0 + t) \leq r \leq (n - 1 + t)$

CISC1110-Spring2012-Raphael-lec4

Sample Code

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

using namespace std;

int main() {
    int iSecret;

    srand ( time(NULL) );
    iSecret = rand() % 10 + 1; // generate secret number

    cout << endl << endl << "Secret number " << iSecret << endl;
    return 0;
}
```

CISC1110-Spring2012-Raphael-lec4