CSCI 123 Introduction to Programming Concepts in C++

Brad Rippe

Intro to the course
Why are you here?
Why are you here?
Welcome to the course

- Brad Rippe

- Email – brippe@fullcoll.edu
  - Add Subject “CSCI123 – topic”

- Phone – (714) 808 – 4872

- Office – No Office (TBD)
CSCI 123 Introduction to Programming Concepts in C++
(Syllabus for Fall 2010 August 16 - December 8)

Instructor: Brad Ripp
Department: Computer Science
Office Hours: Monday, Wednesday, Friday, 9-9 p.m. By appointment — call or email
Lab: 1 hour per week attendance
Office: 118
Phone: (714) 732 - 6064
Email: brippe@fullcoll.edu (Use “CSCI123” as the message subject)
Web Site: http://staff.fullcoll.edu/brippe/csci123/
Twitter: @brippe

Prerequisites:
Math 142 (Introduction) or three years of high school mathematics including trigonometry with grades of "C" or better.

Course Description:
This course is an introduction to the basics of programming using a structured programming language, C++. Students will design algorithms, write external documentation, design and write source code in C++, and write internal documentation for programming problems.

Course Information
Units: 4
CRN: 12883
Course Days: Monday & Wednesday
Course Time: 4:00 - 5:50
Course Room: 621

Textbook (THE BOOK IS REQUIRED)

Problem Solving with C++, 7th Edition
By: Walter Savitch
Publisher: Brooks/Cole
Copyright: 2009
Format: Paper: 1011 pp
Price: $24.0
ISBN-10: 0534385144

Student Evaluation:
The course is based on 3 tests (2 week and comprehensive final), Programming Projects, Assignments, quizzes, lab work and other assignments. There will be no makeup tests given. If you require any special accommodations, you must notify the instructor in a timely manner. Otherwise, you will be expected to adhere to the information included in this course syllabus. The work you do in this class is expected to be YOUR WORK and not a group project unless specified. See the bottom of academic honesty.
The Book

• Do I need the book?
  ○ **YES!!!** You should purchase the book or a version of the book.

**Problem Solving with C++**
7th Edition

By:  *Walter Savitch*
Publisher: Addison-Wesley
The Book

• (YOU) Do I need the book?
  o (ME) **YES!!!** You should purchase the book or some edition of the book.

Problem Solving with C++ - 7th Edition
What's Expected before you take CSCI 123?

• **Prerequisites:**
  Math 142 (Trigonometry) or three years of high school mathematics including trigonometry with grades of "C" or better.
# Grading

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tests</td>
<td>60%</td>
</tr>
<tr>
<td>Programming Projects (2)</td>
<td>20%</td>
</tr>
<tr>
<td>Quizzes</td>
<td>5%</td>
</tr>
<tr>
<td>Homework/Assignments</td>
<td>10%</td>
</tr>
<tr>
<td>Lab Hours (1 hour per week)</td>
<td>5%</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
</tr>
<tr>
<td>Percentage Range</td>
<td>Grade</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------</td>
</tr>
<tr>
<td>90 – 100 %</td>
<td>A</td>
</tr>
<tr>
<td>80 – 89 %</td>
<td>B</td>
</tr>
<tr>
<td>70 – 79 %</td>
<td>C</td>
</tr>
<tr>
<td>60 – 69 %</td>
<td>D</td>
</tr>
<tr>
<td>Below 60 %</td>
<td>F</td>
</tr>
</tbody>
</table>
Calculating Grade

\[
\text{[Your points]}/\text{[Total points]} \times \text{[Weight]}
\]

\[
\begin{align*}
90/100 \times .60 &= .54 \\
100/100 \times .20 &= .20 \\
5/10 \times .10 &= .05 \\
\text{lab} &= .05 \\
\text{hw} &= .05 \\
\text{Total} &= 89\%
\end{align*}
\]

<table>
<thead>
<tr>
<th>Grade Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 Tests</td>
<td>60%</td>
</tr>
<tr>
<td>Programming Projects</td>
<td>20%</td>
</tr>
<tr>
<td>Quizzes</td>
<td>5%</td>
</tr>
<tr>
<td>Homework</td>
<td>10%</td>
</tr>
<tr>
<td>Lab Hours (1 hour per week)</td>
<td>5%</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
</tr>
</tbody>
</table>
Calculating Your Grade - calculateGrade

There is also an application that will calculate your grade when you log into the server.

The application is called calculateGrade.

Type "calculateGrade" from the command prompt.

```
fccsci:/home/brippe# calculateGrade
How many tests have you taken?
```
Checking your Grades Online

- Log into myFC – https://myfc.fullcoll.edu
- Click Gradebook Tab
- Click the course to view grades
Your responsibilities

• IF YOU DO NOT PLAN ON CONTINUING THIS COURSE, IT IS YOUR RESPONSIBILITY TO DROP IT!!!!!

• Student Wait Time for Late Instructors
• Academic Honesty
• ADA – if you have specific need for assistance, please notify your instructor
Course Schedule

• Tentative schedule outlined in the handout

• Check the course web site
  http://staffwww.fullcoll.edu/brippe/csci123
    o Assignments
    o Lectures
    o Handouts
    o Sample Code

• http://staffwww.fullcoll.edu/brippe/csci123/schedule.aspx
The Development Environments

• Popular IDEs/Compilers
  o GNU’s g++ with a text editor of choice (OUR COMPILER)
    • http://gcc.gnu.org/
  o Microsoft’s Visual C++ 2010 Express Edition
    • http://www.microsoft.com/express/download/
  o Borland’s C++ builder
  o MinGW
    • http://www.mingw.org/
  o Dev-C++ (BloodShed)
    • http://www.bloodshed.net/dev/devcpp.html
What environment will we use?

- We will use the Linux operating system (OS) with the GNU g++ compiler.

- You can use any editor you like
  - Linux provides nano, vim, emacs

- You **must compile** your programs and **execute** them on the fccsci.fullcoll.edu server to receive full credit
Text Editors

• **Linux Editors**
  o Nano
    • [http://www.nano-editor.org/](http://www.nano-editor.org/)
  o Vim
  o Emacs
CSCI 123 Introduction to Programming Concepts in C++

Brad Rippe

Hardware and Software
Hardware (computer systems)

- PC (Personal Computer)
- Workstation
- Server
Virtualization
Network
Common Components

- Input Devices
- Output Devices
- CPU (*Central Processing Unit*)
- Main Memory (RAM)
- Secondary Memory (Hard Drives)
Computer Memory

- Main Memory
  - Scratch Paper
    - Temporary

- Secondary Memory
  - Disk Drives, USB Drives
    - Extended life
Main Memory

- (RAM)
  - Bit – binary digit
  - Byte – eight bits
  - Address

![Diagram showing memory addresses and binary values]
The Computers Brain

- CPU
  - What does it do?
Software

- Operating System
  - DOS, Windows, Linux, Unix, VMS, MacOS, and more...

- Applications
  - Console, Web Browser, Office Suite, Email Client, Utilities, and much much more...
Programs/Applications
Languages

- High-Level
- Assembly
- Machine
- Hardware
High Level Languages

- Examples
  - C/C++
  - Java
  - C#
  - Visual Basic

- Look similar to English
- Easier to write than low level language
- Must be translated or interpreted into low level language by the compiler
- Abstracted from the computer hardware

```java
Chat chat = new Chat();
chat.sendMessage("Hello ");
chat.sendMessage("what’s up?");
chat.sendMessage("Did you go to “+ “class today?");
```
Low-Level Languages

• Close to the hardware
• Assembly Language
  o Abstract machine language one level
  o Easier to write than Machine language

• Machine Language
  o CPU understands
  o 100011 00011 01000 00000 00001 000100

; display message 3
  mov  ah, 02h
  mov  bh, 00h
  mov  dx, 0200h
  int  10h
Compilers

- Program which takes your source code and translates it into machine code
- Computer understands the machine code and can execute it on the platform
Linker

- Run after the compiler
- Combines your object code with any of the pre-existing object code (input and output)
- Generates a complete program
Algorithms

• Sequential steps for solving a problem or task
• Language independent
• Written in plain English
• Allows programmers to concentrate on the solution without worrying about the implementation details
Cake Algorithm

• Stir into a large mixing bowl
  o 2 eggs
  o 4 cups of water
  o Cake mix

• Once all the lumps are gone
  o Preheat oven to 400 degrees
  o Place cake mix in a 4X7 greased cake pan
  o Bake for 35 minutes
  o Cool for 15 minutes and serve
Simple Sort Algorithm

Version 1

1. Get a list of unsorted numbers
2. Repeat steps 3 through 6 until the unsorted list is empty
3. Compare the unsorted numbers
4. Select the smallest unsorted number
5. Move this number to the sorted list
6. Remove the selected smallest number from the unsorted list
7. Stop
Simple Sort Algorithm

7 9 2 5 8 6 1
Simple Sort Algorithm

1

7 9 2 5 8 6 1
Simple Sort Algorithm

7 9 2 5 8 6

1 2
Simple Sort Algorithm

7 9 5 8 6
1 2 5
Simple Sort Algorithm

7 9
8 6
1 2 5 6
Simple Sort Algorithm

7 9

1 2 5 6 7

8
Simple Sort Algorithm
Simple Sort Algorithm

Success!!!
Program Design

- Problem Solving Phase
  - Algorithm is design
  - Algorithm is tested

- Implementation Phase
  - Algorithm is translated into C++
Object Oriented Programming

- Objects are designed to simulate real world items
- Objects have characteristics and actions
- Not specific to C++
- C#, PHP, Python, Java and more

- Main Characteristics
  - Encapsulation
  - Inheritance
  - Polymorphism
Intro to C++

- Derived from C

- Developed at AT&T Bell Laboratories by Bjarne Stroustrup in the 80’s

- Overcome some of the shortcomings of C

- Object Oriented was the one of the main features of the language
#include <iostream>
using namespace std;

int main() {
    return 0;
}
```cpp
#include <iostream>
using namespace std;

int main() {
    cout << "Sending text to the screen.\n";
    return 0;
}
```

- `cout` – sends output to the monitor
- "\n" – sends the cursor to the next line (newline)
- `return 0;` - terminates the program
- `<<` - insertion operator
Sample C++ Programs

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

int main() {
    int someInteger = 0;
    cout << "Declaring variables. Type a number.\n";
    cin >> someInteger;
    cout << "Here's your number ";
    cout << someInteger;
    cout << endl;
    return 0;
}
```

- **cin** – sends input from the keyboard to a variable
- **endl** – sends the cursor to the next line (newline)
#include <iostream>
using namespace std;

int main() {

    variableDeclarations

    ...

    statement1

    statement2

    ...

    return 0;

}
Compiling and Running

• C++ source code is text.
• It can be written with any text editor:
  o VIM
  o EMACS
  o Nano
  o More...

• This is old school and still cool!!!
Compile from Command line

• G++ Compiler

  g++ -Wall hello.cpp -o hello

  Run by typing:
  ./hello
Bugs

• Grace Hopper and others found the first actual bug

• Bugs are errors
  • syntax errors (compilation time)
  • runtime errors
  • logic errors
Resources

• Course Web Site
  o [http://staffwww.fullcoll.edu/brippe/csci123](http://staffwww.fullcoll.edu/brippe/csci123)

• Course Style Guide
  o [http://staffwww.fullcoll.edu/brippe/csci123/style.aspx](http://staffwww.fullcoll.edu/brippe/csci123/style.aspx)

• Course Syllabus
  o [http://staffwww.fullcoll.edu/brippe/csci123/syllabus.aspx](http://staffwww.fullcoll.edu/brippe/csci123/syllabus.aspx)

• Course Schedule (tentative)
  o [http://staffwww.fullcoll.edu/brippe/csci123/schedule.aspx](http://staffwww.fullcoll.edu/brippe/csci123/schedule.aspx)