

- Day and time: **Wed. 6:30-9:00 PM**
- Room: **STH 135**
- Instructor: **Bill Bonnice** Office: Regis Annex, top floor
- Home phone: 603-888-4807 Office phone: 603-897-8540
- Home e-mail: cf1bb@aol.com E-mail: wbonnice@Rivier.edu

Office hours: Please let me know whenever you need help. Many ideas in this class will be new to you and you will need to study them beyond the classroom in order to understand them fully. You are invited to stop by my office whenever I am there or make an appointment for a different time. I will be happy to remain after class to work with anyone and I will also respond to e-mail questions. My scheduled office hours are:

Monday 3:00 – 5:00, Wednesday 4:00 – 5:00, Friday 12:30 – 1:30
and by appointment(Please feel free to call me at home or at my office.).

Course Description: An introduction to problem solving and algorithmic solutions using a modern object-oriented programming language such as Java. Topics include: primitive types, control structures, and simple input/output operations. Emphasis on functional abstraction and recursion, and on user-defined types, data structures, and run-time storage management.

Objectives: In this course students will understand the object-oriented approach for designing and implementing software solutions.
Master fundamental programming skills.
Learn problem solving techniques.

Course Material: The course will cover ---

The concept of class and how classes can be used to write programs.
Fundamental control structures, such as conditionals and loops.
Class and method design: instance data, visibility, scope, parameter passing, object references, nested classes.
Arrays and vectors.
Class derivation, class hierarchies, overriding, and polymorphism.
Java I/O classes.
Applets
Graphical User Interfaces(GUIs)

Textbook:

Object-Oriented Problem Solving, Java, Java, Java, 2nd Edition
Ralph Morelli
Prentice-Hall, 2003

Supplementary materials are available on line at <http://www.prenhall.com/morelli>.

Course Work:

We will cover approximately one chapter per week. Since this is a second course in java, when we get to the chapters that simply describe the java language, we will do two chapters a week on that material. This will be a quick review of that material.

Participation:

Students are encouraged to participate in class discussions conducted both in- and outside the classroom. An e-mail group has been set up to facilitate this and, with the permission of those included, a list of phone numbers and e-mail addresses of the students in the class will be circulated.

Reading and Homework Assignments:

Students are required to read thoroughly all the assigned readings. The reading and homework assignments are essential to understanding and learning the material presented in class, and to writing the programming assignments and demonstrating mastery of the course material. You must turn in your homework assignments at the next class after they were assigned.

Programming Assignments:

As a rule there will be a programming assignment due each week. In addition, there will be one or significant programming projects. Students will be required to present a description of these projects to the class.

Although collaboration is encouraged to discuss assignment specifications, language constructs and Java library features, and conceptual aspects of the solution design, the programming assignments must be entirely your own work.

Submit a hard copy with all the Java source files that you wrote (Do not submit library or textbook Java classes). Include a copy of the output from running your program with *several* sets of test data, if data is input. You may paste the output as a comment at the end of your program.

Deadlines for turning in the programming assignments are extremely important. No late submission are accepted, unless you provide proof that serious medical, personal, or family circumstances prevented you from meeting the deadline. Extensions may be requested only in exceptional cases, approved by the instructor, and will be granted if the request is at least three days prior to the deadline. If you want to receive prompt feedback, start early on writing the program, participate in class and e-mail discussions, and always submit your assignment on time. If your program does not compile, together with your program, submit the first 15-20 lines of compilation errors.

Make sure that all the source files are properly documented with comments. It is essential to include a block comment at the beginning of each file that has: file name (with extension), author name, programming assignment number, short description, history information: creation date and modification dates with short descriptions about the nature of the code changes and/or additions.

Examinations:

There will be several in-class tests and a final examination. Except for the multiple-choice part of the final exam, usually examinations are open-text, open-notes. **No collaboration is allowed while taking the tests.** On some tests you will be required to write and run a Java application or applet on your own, so it is important that you use the weekly programming assignments to develop that skill.

Grading: Final grades are broken up as follows:

Non-programming Homework Exercises	15%
Programs	30%
Project	10%
Tests	20%
Final Exam	20% (10% for Programming part, 10% for question and answer part)
Self-evaluation	5%

Program Submission Guidelines

Submission of programming assignments and of homework assignments that require you to write Java programs must comply with the following guidelines:

1. Insert a block comment at the beginning of the .java file, in which you provide the following information:
 - **Name of the file**, with extension, as it appears in the filesystem of the machine on which you did the program development
 - **Your name**, as author of the file
 - The **date** when the file was created
 - If homework assignment, **correct name and number of the homework assignment** to which the program belongs, with information about the **programming project number and page** in the text with which the homework assignment is associated.
 - If programming assignment, **correct name and number of programming assignment** to which the program belongs
 - Brief **description of the program** to which the file belongs
 - Optional: Dates when the program was modified with short description of the changes/additions.

Examples:

1. For example, the block comment given below gives information about the file ConvertTime.java that has the Java program that implements programming project 2.6 (page 105) which was assigned as an extra credit part of Program # 3.

```
// File:          ConvertTime.java
// Author:       Mihaela Sabin
// Created:      1/9/05
// Program # 3   Programming Project 2.6, page 105
//              Convert seconds into hours, minutes, seconds
// Modified:     1/15/05 complete conversion
```

2. Another example shows the block comment for Programming Project 2.11, page 106:

```
// File: MilesPerGallon.java
// Author: Mihaela Sabin
// Date: 1/9/05
// Programming Project 2.11, page 106
// Calculate miles per gallon of gas for a trip
```

3. Provide an 'end' comment for each class and method in your program. For example:

```
public class SomeExample
{
    public static void main(String[ ] args)
    {
        System.out.println("Just an example...");
    } // end main( )
} // end SomeExample
```

TERM PROJECT PROGRAM

You may do this project alone or with at most one other person.

IMPORTANCE OF GOOD COMMENTING: Write this program as if your boss knows nothing about programming and you want him to understand it. Thus you must put in sufficient comments to clarify anything he might not understand. On the other hand, commenting things which are clear clutters up a program so comment judiciously. This will count one fifth of your grade on this program.

If you would like to dream up your own project, submit a detailed proposal to me for my approval. Otherwise choose one of the following programs as a project:

Go to the Morelli website(www.prenhall.com/Morelli), click on the Companion Website and on the page that comes up, click on the author's website, and then click on [Sample Syllabus \(Trinity Spring 2002\)](#). In that syllabus there are links to the following, which you may choose from as projects:

- I. Lab 3 : The Invoice Class
- II. Lab 8: The Actuary Problem
- III. Lab 9: Newton's Method for Finding Square Roots
- IV. Lab 10: A Name Game
- V. Lab 11: Statistics

VI. Programming Assignment # 1 (In the r. h. col. of week 6): Ralph Walde's number game applet.

VII. Programming Assignment # 2 (In the r. h. col. of week 10): The Sliding Tile Puzzle.

VIII. Simulate an ATM machine with an applet. (Not from Morelli. Create your own description.)

Schedule:

Wed. 2 / 23 : Turn in which project you are going to do and your partner's name.

If you are going to work alone, say so.

Wed. 3 / 16 : Turn in UMLs and a written plan or pseudo-code for your program together with a schedule of times you will work on the project. Include statements of what you want to accomplish by what date plus time for write-up and time to prepare to present your results to the class at our last meeting, Apr. 23, before the final, which will be on Apr. 30.

Wed. 4 / 13: Turn in your completed code with output from running it.

Wed. 4 / 27(or earlier): Make a thorough and understandable presentation of your problem and your solution to the class. You will have about 15 minutes and you and your partner should share equally in the presentation time. The grade of a partner who does not do her/his share of the presentation will suffer.

EXTRA CREDIT: If you get ready early and make your presentation ahead of time at the next to last class on Wed., Apr. 13, you will get an extra 10% on your project grade.

EFFECT ON FINAL GRADE: From the syllabus, you will see that the Project will constitute 10 % of your final grade so it behooves you to do a good job on it!

Below I have taken directions that Morelli wrote for one of the projects and modified them slightly.

They spell out in detail the ground rules for doing a project. Note that the tone is that you should work mainly on your own. I have a different philosophy. My main goal is that you **learn**. I encourage you to help one another learn. The important thing, as far as I'm concerned, is that you learn it well enough to produce the final product on your own; that the final product is your *own* work and not copied from someone else or from another source. **If, during the presentation, I call on you to go to the board and reproduce part of your work you should be able to do so!**

READ THIS CAREFULLY!

This is a two-person team programming project but each individual will receive a separate grade on the assignment depending on the work that each contributes to the final result. (Of course, if you choose to work individually, you may do so.)

Each team must clearly document, in the header of their main applet file, what each team member contributed to the solution. Hopefully, each team member will contribute equally to the final product. But, if not, you must clearly describe who did what. As in the previous assignment, you may not copy code from anybody else, either in the course, or from a friend, or off the web, and so on. To do so, would constitute **plagiarism** and will be dealt with as such.

These rules do not mean that you cannot talk to other classmates about your work. It is okay to discuss in general how you are tackling a particular problem -- e.g., "I used an array to store the buttons in the key pad" or "I defined a method to do this or that task". You may also give or receive help from a classmate about a particular error message or program bug. Also, if you need help with either design or programming, you may consult with friends, the Instructor or TAs (Note that there are TAs. Their office hours and places are posted near the STH computer labs.)

If you receive substantial help from a friend, a TA or Instructor, you must document it in your program's header by providing the name of the person who helped you and a brief description of the help you received. (This is similar to a citation to a reference when writing a term paper.)

HAVE FUN !!!

CS 181 Self-evaluation

This Self-evaluation is due on W 4/27/05, our last class before the final exam. It will count as 5 per cent of your final grade.

You will be scored on the quality of your evaluation of yourself, your work in this course and your progress and understanding. In other words you could say that you deserve a low grade in the course and give good reasons for that and receive maximum credit on the Self-evaluation.

On the other hand, you could say that you deserve an "A" in the course and receive a low score on the Self-evaluation because you did a poor job justifying your claim for an "A".

1) Note: Grades are based on understanding, progress and achievement, not on effort. It usually takes a lot of effort to achieve understanding and progress in CS 181 but effort alone, without achieving understanding and progress, does not warrant a good grade. See 6) below.

In addition to stating what grade you think you deserve in the course, along with justification, you should evaluate yourself on the following:

2) How well did I work with my classmates in and out of class? **Although I may have received help in developing my weekly assigned programs, was I able independently to complete the program which I finally submitted?**

3) How much did I work outside of class and how was the quality of that work? Did I study the text thoroughly with a pencil and paper at hand to try to work out examples and try to predict output of programs before running them? Did I try to work out things that I didn't understand? Did I write down questions to be answered in class? How well did I do on the homework and how much time did I put in on it? What was the quality of my effort?

4) QUIZZES and TESTS: How did I do on them? How well did I prepare for them? Afterward did I consult with my classmates about them and compare my answers with theirs?

5) IN-CLASS PARTICIPATION: Did I ask questions in class? How much did I contribute to class discussion?

6) UNDERSTANDING, PROGRESS, and ACHIEVEMENT: How much did I learn and understand in this course? Did I realize my *VISION* for myself? Did I meet my expectations? How much progress did I make?

(Compare where you are now to where you were at the beginning of the course. Has there been much change?) What have you achieved in this course?

7) CRITIQUE OF THE PROFESSOR: What did you like least about my teaching? What did you like most? Suggest at least one thing that I could do to improve my teaching.

8) CRITIQUE OF THE COURSE: What did you like least about the course? What did you like most? Suggest at least one thing that could be done to improve the course.

9) MISCELLANEOUS: Here say anything else you'd like to say.