

**VIRGINIA STATE UNIVERSITY**  
**School of Engineering, Science and Technology**  
Department of Mathematics and Computer Science  
CSCI-260 Object-Oriented Programming      Spring 04  
  
Course Syllabus

Section 01	Meets: MWF 10:00-10:50 HM 18E
Instructor: Dr. Cecil Morris	Office HM 308 Telephone: 524-5413
e-mail: cmorris@vsu.edu	Course webpage:
Office Hours:	MWF 8:00-10:00, 11:00-12:00; TR 8:00-9:30

### **COURSE DESCRIPTION**

This course introduces software design methodology using C++. Students are introduced to objects and methods, the object-oriented features of C++. It includes the following topics: data abstraction, hiding and encapsulation, inheritance and derived classes, static and dynamic binding, polymorphism and virtual functions.

### **PREREQUISITES:**

Programming in C++ II (CSCI-250) with grade at least C. This prerequisite will be strictly enforced.

### **COURSE TEXT**

The following text is required:

*Bronson, Gary J. Program Development and Design Using C++, Brooks Cole Publishing Company, 2000*

As supplementary materials, the student must acquire a few high density floppies. These must be submitted with your projects. See document: "Guidelines for Project Documentation", for more details on how to submit projects.

### **LEARNING OUTCOMES, ACTIVITIES AND EVALUATION PROCEDURES**

The different evaluation methods serve to assess the learning outcomes. The student is expected to have acquired the following:

### **KNOWLEDGE/SKILLS**

1. Write simple recursive functions.
2. Understand the relation between pointer and arrays/structures and functions.
3. Define concepts of class, objects, methods and their C++ implementation.
4. Understand the basic ideas of object-oriented programming: data abstraction, information hiding, and encapsulation, with their associated syntax.
5. Demonstrate the use of classes in simple programs.

6. Demonstrate the use of the basic standard classes for I/O

## ABILITIES

1. Identify the correct use data abstraction, information hiding, and encapsulation in a wide range of applications.
2. Write applications that use and polymorphism and virtual functions, clearly understanding static vs dynamic binding.
3. Write programs that take advantage of the class hierarchy and inheritance properties of classes.

## EVALUATION

At the instructor discretion, quizzes and assignments (projects) will be given at regular intervals, typically three of each. The instructor may encourage class participation by providing extra credit points to be used toward quizzes. Practice homework will be provided weekly.

There will be two exams, comprehensive and coordinated (prepared by the Department): midterm and a final. The numerical grade is computed as follows:

$$\begin{aligned} \text{MA} &= (2/3) \text{QA1} + (1/3) \text{ME} \\ \text{FG} &= (1/3) \text{MA} + (1/3) \text{QA2} + (1/3) \text{FE} \end{aligned}$$

where

QA1 : average of assignments and quizzes before Midterm

QA2 : average of assignments and quizzes after Midterm

ME: Midterm examination

MA: Midterm grade (reported)

FE: Final Examination

FA: Final grade.

## COURSE REQUIREMENTS

This is a continuation of the course CSCI-250 (Programming in C++ II). Students are assumed to have working knowledge of user-defined functions with a variety of parameter types. It is important as well that the student be familiar with pointers and their relations with arrays and structures.

In order to achieve success in the course it is fundamental the abilities acquired by doing working on projects and the weekly assigned homeworks. The texbook sample problems are available in your classroom computer and often will be used in class practice. Feel free to copy the programs to your computer. Students should be prepared to spend 4 to 6 hours per week *besides* time in the classroom.

Quizzes given at regular intervals will should serve as self-check and are often indicators of future performance in exams. Programming projects will be assigned approximately every two or three weeks and may consist of two to three problems, to be submitted following the instructions in the **“Guidelines for Project Documentations”**. Quizzes will be given every three weeks more or less.

Late projects will in general not be accepted, or they will be penalized at the discretion of the instructor. Make-ups in general will not be given, unless extenuating circumstances. Here again, the policies may vary from instructor to instructor. Make sure you check with him or her regarding the policies.

Students are expected to:

1. Attend class regularly and on time.
2. Turn in all assignments on time (refer to GUIDELINES)
3. Read carefully the relevant portions of the text.
4. Conduct themselves in a quiet and respectful manner during lectures.
5. Study thoroughly both theory and practice.
6. Seek the professors, if needed, during office hours.
7. Be honest and do your own work.

## **GRADING STANDARDS**

Grading and treatment of academic dishonesty are conformed to University rules. Please read carefully the attached excerpts from the Catalog and the Students Handbook.

## **BIBLIOGRAPHY/READING LIST**

For a good treatment of Object Oriented programming, possibly using other languages see:

- Slack, James. *Programming and Problem Solving with Java*, Bruce-Cole 2000.
- Dale, N. Weems, C. and Headington M.: *Programming and Problems Solving with C++* Jones and Bartlett, 1998
- Lippman, S.B. *C++ Primer* 2nd Edition (more advanced) Addison-Wesley, 1993
- De Champeaux, Dennis, Lea, D and Faure, P. *Object-Oriented System Development* Addison-Wesley, 1993