

**CS321, Programming Languages**  
**Spring Term, 2008 (3 credits)**  
**Section 1, MWF 9:30-10:20a.m. Comm101**

---

INSTRUCTOR: Fr. Boniface Hicks, OSB  
OFFICE: 101 Physics Building  
PHONE: 724-805-2143 or x2143  
EMAIL: [boniface.hicks@email.stvincent.edu](mailto:boniface.hicks@email.stvincent.edu)  
IM: frdrboniface (AIM)

OFFICE HOURS: MF 1-3pm

And by appointment  
and "drop-in".

---

**COURSE DESCRIPTION:**

This course covers topics in programming languages. The student will explore programming language features and programming paradigms. Course topics cover programming language translation issues including syntax, types and semantics; programming paradigms, including imperative, object-oriented, functional and logic programming, and special topics, including event handling, concurrency and correctness, as time allows. The course will cover various languages as examples and topics of study, but Java will be a primary example throughout.

**WHY TAKE THIS COURSE?**

A course on programming languages is considered essential to a computer science education by the Association of Computing Machinery (ACM) as described in their outline of computer science curricula:  
"A programming language is a programmer's principal interface with the computer. More than just knowing how to program in a single language, programmers need to understand the different styles of programming promoted by different languages. In their professional life, they will be working with many different languages and styles at once, and will encounter many different languages over the course of their careers. Understanding the variety of programming languages and the design tradeoffs between the different programming paradigms makes it much easier to master new languages quickly. Understanding the pragmatic aspects of programming languages also requires a basic knowledge of programming language translation and runtime features such as storage allocation."

**PREREQUISITE/CO-REQUISITE**

To learn about programming languages in general, it is necessary to have a thorough and comfortable understanding of at least one language. The requirement of CS 111 presumes that students are familiar with C++. Furthermore, to understand how languages translate human ideas to codes executable by the computer, it is important that students have some understanding of low-level languages such as assembler, captured by including CS 230 as a pre-requisite.

**CORE GOALS**

This course contributes especially toward the following core curriculum goals, listed in order of emphasis:

- To form habits of ordered inquiry, logical thinking, and critical analysis
- To develop mathematical skills and quantitative literacy
- To develop effective communication skills

**CIS DEPARTMENT GOALS**

This course contributes to the following departmental goals, listed in order of emphasis. A minimum of software engineering is emphasized in this course, as this is not a software development course. Problem-solving, however, is key to this course.

- The CIS graduate should demonstrate the ability to manage the complexity of a technical problem through the use of good problem solving skills and software engineering skills, as well as ethical and decision-making skills.
- The CIS graduate should have a broad knowledge of the field of computing.

**COURSE GOALS**

Specific course goals include the following. These goals will be assessed through the use of assignments, projects, labs, and tests.

- By the end of the course, the student should be able to identify the impact of different features (strong typing, memory management, dynamic dispatch, exception handling, etc.) on criteria of computer time/space efficiency, programmer time efficiency, program safety and power of expression.
- By the end of the course, the student should be able to understand language syntax, features and vocabulary sufficient to read a language specification and answer questions about a particular program's behavior.
- By the end of the course, the student should be able to use programming language tools for producing clearer, safer programs.
- By the end of the course, the student should be proficient in Java.
- By the end of the course, the student should be able to choose between languages and paradigms to solve a particular problem based on criteria of computer time/space efficiency, programmer time efficiency, program safety and power of expression.
- By the end of the course, the student should be able to quickly learn a new programming language.

These goals will be assessed mainly through the use of homework assignments, in-class work, quizzes and exams.

## CS310, Programming Languages, Spring 2008

TEXT: *Programming Languages: Principles and Paradigms* by Allen B. Tucker and Robert E. Noonan, 2/e

We will follow the text for most of the course, but some supplemental materials will also be used to broaden and deepen the material covered.

### ADDITIONAL READINGS:

Other readings may be assigned and will be available from the instructor, online or in the library. The SVC Blackboard site will be used for this class. It may be accessed at <http://cms.stvincent.edu>.

### GRADING:

30% Midterm Exam

30% Final Exam

40% Projects, Labs, and Other Assignments (homework)

### GRADING SCALE:

The final grading scale may be curved at the discretion of the instructor, but in order to monitor your progress during the term, you should assume the SVC grading scale as printed in the College Bulletin. Curved grades will result in a higher letter grade than that achieved solely by the percentages above. For example, an earned 'B' may become a 'B+'. The curve depends upon the performance of the class as a whole as well as individual in-class participation and perceived effort. NOTE: **If a student does not attain a passing average in the exam category, that student will receive a failing grade for the course.**

### EXAMS:

There will be two exams, which, in total, will account for sixty percent of the final grade. Exams may cover any topics discussed in class, in the text, in any additional readings and in the assignments. Exams must be taken when scheduled. No books, notes, electronic devices or other assistance (including other people) are to be used during exams **except** you may use **one two-sided page of handwritten notes**. Cell phones, pagers and PDAs must be turned off during exams. If you have a need to keep a cell phone or pager on during class, please let me know. **Once an exam has begun, you are not permitted to leave the room until you hand in the exam.** If any students miss a test due to verifiable extreme circumstances, the final will be used as the grade for that exam. (See the Section labeled "DEFINITION OF EXTREME VERIFIABLE CIRCUMSTANCES".)

### COURSE POLICIES:

Programming involves typing code into a source file, compiling it, testing it and fixing it, if necessary. **For all programming assignments, a hardcopy of the commented program will be turned in and the source code placed on the appropriate CIS dept. network drive or submitted according to the professor's instructions.** Any other assignments should be word-processed if at all possible. If handwritten, the handwriting must be clear and legible. **Late assignments will not receive full credit** since the solutions may be discussed in class the day that the assignments are due. The percent deducted depends upon how late it is turned in. Do not turn in any exercise or assignment on paper torn from spiral-bound notebooks or on any size paper other than 8.5x11 inch. **Multiple pages must be stapled or paper clipped together or points will be deducted from the assignment grade. For all assignments and exams, illegible answers will not be graded and no points will be awarded.**

**If you cannot attend a class:** Assignments may be placed in the appropriate network drive or emailed to me if you cannot attend a class, but will not be graded until a paper copy is submitted. The emailed assignment must be received before the start of the class when the assignment is due. Program source code should be submitted as plain text. The paper copy must be submitted by the next class period or no points will be awarded for that assignment. Assignments should be emailed only in case of some emergency and emailed assignments will not be accepted more than twice a term. Students who are **participating in sports** are expected to follow the College Handbook procedures for excused absences from class and exams.

**Students with disabilities** who may be eligible for academic accommodations and support services should please contact Mrs. Sandy Quinlivan by phone (724-805-2371), email ([sandy.quinlivan@email.stvincent.edu](mailto:sandy.quinlivan@email.stvincent.edu)) or by appointment (Academic Affairs-directly above the post office). Reasonable accommodations do not alter the essential elements of any course, program or activity.

**If assignments from two or more students are unduly similar**, they will be considered to be in violation of the SVC Academic Honesty Policy, and will be treated as such under the guidelines of that policy. "Unduly similar" in this case means that a prudent individual would reasonably conclude that the assignments were written/completed by the same person. The assignment grades for those students involved will be severely penalized **and the incident will be reported to the Academic Deans Office.** A second incident will result in failure of the course.

**Class attendance** is most strongly recommended. Some classes will include in-class assignments, quizzes or computer lab work for which points will be awarded. **If these are missed due to an unexcused absence, they cannot be made up.** Additional points may be added to the final grade based upon class participation. If you miss a class, it is your responsibility to get any notes, handouts and assignments. If a medical or sport excuse is provided, you will be given the opportunity to make up missed in-class assignments or quizzes. **After 1 weeks worth of unexcused absences, 2 percentage points will be deducted from your final grade points for each additional missed class.**

If a class or office hours must be canceled for any reason, I will try to contact you by email via Blackboard as soon as I know of the cancellation and post the cancellation on Blackboard. Therefore, you should check your profile on Blackboard and make sure the email address is one you read frequently. If assignments are due when a class has been canceled, they may be turned in at the next class without penalty.

Students should consult the CIS Department Policies webpage (<http://cis.stvincent.edu/policies.html>) for additional information regarding course and department policies.

#### DEFINITION OF EXTREME VERIFIABLE CIRCUMSTANCES

Examples of extreme circumstances are serious illnesses or the death of a family member. Examples of non-extreme circumstances are nonrefundable airline tickets, sporting events and concerts. Unfortunately, I have encountered a number of attempts at deception, so proof of the extreme circumstance will be required, such as a note from a nurse, doctor or coach, or an obituary notice, or a receipt from a car-towing company, etc.

#### TOPICS

##### 0. Learning Java

- basic Java programming
- using Eclipse
- reading/tracing/debugging Java code

##### 1. Syntax

- EBNF, Compilers/interpreters, linking syntax/semantics
- syntax of Clite

##### 2. Names

- Scope, lifetime, overloading, symbol table

##### 3. Types

- static/dynamic typing
- basic, non-basic, programmer-defined, and recursive data types
- subtypes, polymorphism and generics
- type system for Clite

##### 4. Semantics

- expressions, state, assignments, control flow, I/O, exceptions
- semantics of Clite, formal semantics

##### 5. Functions

- parameter passing
- recursive functions
- activation records
- runtime stack
- Clite function declaration, function call, semantics

##### 6. Programming paradigms

- imperative basics
- object-oriented programming
- logic programming
- scripting languages

##### 7. Special topics

- event handling (GUI programming)
- Concurrency (threads, synchronization mechanisms)
- Program correctness
- language-based security