RARITAN VALLEY COMMUNITY COLLEGE
ACADEMIC COURSE OUTLINE

CISY 105 – Foundations of Computer Science

I. Basic Course Information

A. Course Number and Title: **CISY-105, Foundations of Computer Science**

B. New of Modified Course: **Modified**

C. Date of Proposal: Semester: **Spring** Year: **2008**

D. Sponsoring Department: **Computer Science (CS) Department**

E. Semester Credit Hours: **4**

F. Weekly Contact Hours: Lecture: **3**
Lab: **2**

G. Prerequisite: **Math 113 - Precalculus II** or **Math 114H - Precalculus Honors**

H. Laboratory Fees: **Yes, at current rate**

I. Department Chair: **Dr. John Sullivan: jsulliva@raritanval.edu**

II. Catalog Description

(Prerequisite: Math 113 – Precalculus II or Math 114H – Precalculus Honors) This is the first course in Computer Science for transfer students. The programming language Java will be taught in conjunction with problem solving methods, algorithm development, and object-oriented design. Topics include data types, control structures, classes, objects, methods, file processing, recursion, and introduction to data structures.

III. Statement of Course Need

This is the first course in the Computer Science transfer program. Over the past few years, enrollment in this course has been steady; this reflects the need for this course as well as the need for the transfer program. Students in this program have transferred to Rutgers, NJIT, The College of New Jersey, and other colleges and universities in and outside New Jersey.

IV. Place of Course in College Curriculum

A. Free Elective

B. Required course for the Computer Science Transfer Program, A.S. degree

C. CIS Programming Elective

D. Transferability: This course transfers to most 4-year Institutions in NJ in Computer Science

V. Outline of Course Content

This course explores the following topics:

1. Introduction to Computers and Java
   - Why Program?
   - Computer Systems: Hardware and Software
   - Programming Languages
   - What Is a Program Made Of?
   - The Programming Process
   - Object-Oriented Programming
   - Algorithms & Problem Solving

2. Understanding Program Structure

3. Java Fundamentals
   - 2.1 The Parts of a Java Program
   - 2.2 The print and println Methods, and the Java API
   - 2.3 Variables and Literals
   - 2.4 Primitive Data Types
   - 2.5 Arithmetic Operators
   - 2.6 Combined Assignment Operators
   - 2.7 Conversion Between Primitive Data Types
   - 2.8 Creating Named Constants with final
   - 2.9 The String Class
   - 2.10 Scope
   - 2.11 Comments
   - 2.12 Programming Style
   - 2.13 Reading Keyboard Input
   - 2.14 Dialog Boxes
   - 2.15 Common Errors to Avoid
   - Number System (Binary, Hexadecimal, 2's complement)

4. Decision Structures
   - 3.1 The if Statement

- 3.2 The if-else Statement
- 3.3 The if-else-if Statement
- 3.4 Nested if Statements
- 3.5 Logical Operators
- 3.6 Comparing String Objects
- 3.7 More about Variable Declaration and Scope
- 3.8 The Conditional Operator (Optional)
- 3.9 The switch Statement
- 3.10 Creating Objects with the DecimalFormat Class
- 3.11 The printf Method (Optional)
- 3.12 Common Errors to Avoid

5. Loops and Files

- 4.1 The Increment and Decrement Operators
- 4.2 The while Loop
- 4.3 Using the while Loop for Input Validation
- 4.4 The do-while Loop
- 4.5 The for Loop
- 4.6 Running Totals and Sentinel Values
- 4.7 Nested Loops
- 4.9 Deciding Which Loop to Use
- 4.10 Introduction to File Input and Output
- 4.11 The Random Class
- 4.12 Common Errors to Avoid

6. Methods

- 5.1 Introduction to Methods
- 5.2 Passing Arguments to a Method
- 5.3 More About Local Variables
- 5.4 Returning a Value from a Method
- 5.5 Problem Solving with Methods
- 5.6 Common Errors to Avoid

7. A First Look at Classes

- 6.1 Classes and Objects
- 6.2 Instance Fields and Methods
- 6.3 Constructors
- 6.4 Overloading Methods and Constructors
- 6.5 Scope of Instance Fields
- 6.6 Packages and import Statements
- 6.7 Focus on Object-Oriented Design: Finding the Classes and Their Responsibilities
- 6.8 Common Errors to Avoid

8. A First Look at GUI Applications
- 7.1 Introduction
- 7.2 Creating Windows
- 7.3 Equipping GUI Classes with a main Method
- 7.8 Using Console Output to Debug a GUI Application
- 7.9 Common Errors to Avoid

9. Arrays
- 8.1 Introduction to Arrays
- 8.2 Processing Array Contents
- 8.3 Passing Arrays as Arguments to Methods
- 8.4 Some Useful Array Algorithms and Operations
- 8.5 Returning Arrays from Methods
- 8.6 String Arrays
- 8.7 Arrays of Objects
- 8.9 Two-Dimensional Arrays
- 8.10 Arrays with Three or More Dimensions
- 8.14 Common Errors to Avoid

10. A Second Look at Classes and Objects
- 9.1 Static Class Members
- 9.2 Passing Objects as Arguments to Methods
- 9.3 Returning Objects from Methods
- 9.4 The toString Method
- 9.5 Writing an equals Method
- 9.6 Methods that Copy Objects
- 9.7 Aggregation
- 9.8 The this Reference Variable
- 9.9 Enumerated Types
- 9.10 Garbage Collection
- 9.11 Focus on Object-Oriented Design: Class Collaborations
- 9.12 Common Errors to Avoid

11. Text Processing and Wrapper Classes
- 10.1 Introduction to Wrapper Classes
- 10.2 Character Testing and Conversion with the Character Class
- 10.3 More String Methods
- 10.6 Wrapper Classes for the Numeric Data Types
- 10.8 Common Errors to Avoid

12. Inheritance
- 11.1 What Is Inheritance?
- 11.2 Calling the Superclass Constructor
- 11.3 Overriding Superclass Methods

- 11.4 Protected Members
- 11.5 Chains of Inheritance
- 11.6 The Object Class
- 11.7 Polymorphism
- 11.8 Abstract Classes and Abstract Methods
- 11.9 Interfaces
- 11.10 Common Errors to Avoid

13. Exceptions and More about Stream I/O

- 12.1 Handling Exceptions
- 12.2 Throwing Exceptions
- 12.3 More about Input/Output Streams
- 12.4 Advanced Topics: Binary Files, Random Access Files, and Object Serialization
- 12.5 Common Errors to Avoid

14. Applets and More

- 14.1 Introduction to Applets
- 14.2 A Brief Introduction to HTML
- 14.3 Creating Applets with Swing
- 14.4 Using AWT for Portability
- 14.5 Drawing Shapes
- 14.6 Handling Mouse Events
- 14.7 Timer Objects
- 14.8 Playing Audio
- 14.9 Common Errors to Avoid

15. Recursion

- 15.1 Introduction to Recursion
- 15.2 Solving Problems with Recursion
- 15.3 Examples of Recursive Methods
- 15.4 A Recursive Binary Search Method
- 15.5 The Towers of Hanoi
- 15.6 Common Errors to Avoid

VI. Educational Goals and Learning Outcomes

A. Education Goals

At the completion of this course, the student will be able to:

1. identify changes in the technology that involves learning sufficient foundational material to enable one to acquire new skills independently after one's formal education is complete. (G.E. 2)
2. investigate technological tools for research, information analysis, problem solving, decision making, quantitative analysis while recognizing, and evaluating ethical issues. (G.E. 3, 5)
3. assess information resources, including print and electronic library resources, as well as the Internet that enables students to use computer software applications, and enhance self-directed learning. (G.E. 3)
4. construct quantitative reasoning techniques to interpret data and solve problems (G.E. 7)

B. Learning Outcomes

At the conclusion of the course, students will be able to:

1. Describe concepts of object-oriented programming such as encapsulation, inheritance, and polymorphism.
2. Solve problems involving decisions and repetition.
3. Develop algorithms that may be used in coding programs.
4. Apply algorithms such as the sequential sort and search in solving problems.
5. Design classes using appropriate *instance data* and *methods*.
6. Design programs using traditional structured as well as object oriented methods.
7. Code programs using the correct syntax of the programming language.
8. Understand where testing fits in the software development process.
9. Test programs adequately with appropriate input data.
10. Distinguish among programming errors (syntax, logic, run-time)
11. Correct programming errors.
12. Understand the difference between a Java application and a Java Applet.
13. Gain some familiarity with HTML (Hypertext Markup Language)

VII. Modes of Teaching and Learning

A. Lecture/Discussion - Lecture with demonstration of   problem solving techniques using Java Programming Language.

B. Laboratory – Lab time to analyze, design, develop, test, debug, and document Computer Programs using documentation tools such as Pseudocode, Unified Modeling Language (UML).

VIII. Papers, Examinations, and other Assessment Instruments

A. Computer Labs – In-class and out of class assignments
B. Quizzes
C. Final Examination
D. Other – Weekly Homework assignments

IX. Grade Determinants

A. Lab Projects
B. Homework Assignments
C. Quizzes
D. Final Exam

X. Text and Materials

Suggested Textbook - Gaddis,Tony. *Starting Out with JAVA: From Control Structures through Objects, Third Edition*. Addison Wesley 2008

(Note: The course outline is intended only as a guide to course content and resources. Do **NOT** purchase textbooks based on this outline. The RVCC Bookstore is the sole resource for the most up-to-date information about textbooks.)

XI. Resources

A. Computer access
B. Web Browser Software that is JavaScript enabled
C. JDK 1.5 Software, Textpad Editor
D. Microsoft Visual Studio

Other Resources from Library and Web

A. Java Coding Style Guide: http://www.cwu.edu/~gellenbe/javastyle/references.html
B. Java How to Program, 6th Ed., by Deitel & Deitel, 2004, Prentice Hall. http://www.deitel.com
C. An Introduction to Object Oriented Programming with Java, Thomas Wu, McGraw Hill, 2004
D. Java Program Design 1.5, James Cohoon & Jack Davidson, McGraw Hill, 2004
E. Foundations of Computer Science: From Data Manipulation to Theory of Computation, Behrouz Forouzan, Brooks Cole, 2002
F. Java 2. A Beginner's Guide. 2nd ed. Herbert Schildt, Osborne, 2003.
G. http://www.jguru.com
H. http://www.javaworld.com

I. http://java.sun.com

J. http://java.sun.com/j2se/5.0/docs/api, Java Documentation (API)