

Spectrum Program

Session 1

Course Title: Introduction to Computer Programming Honors: JAVA

Course Description

This course introduces students to structured computer programming using the Java programming language. Utilizing a PC-compatible computer, a Java compiler, and the logical and problem-solving capabilities of Java, students investigate mathematical concepts. They also explore progressively more sophisticated mathematical ideas drawn from number theory, statistics and probability, and other areas of mathematics.

Students will learn about flow of control, fundamental data types, variables, arithmetic expressions, classes, objects, constructors, fields, methods, inheritance, polymorphism, one and two dimensional arrays, and various forms of input and output including files (persistent data). Many of the programming problems that the students will investigate require the understanding of mathematical concepts. Students' computers will be installed with the Netbeans integrated development environments and the latest Sun Java Development Kit. These tools will be used to explore and build Java applications and applets. The students will also learn how computers, technology and the Open Source community impact how we develop and use technology.

This class prepares students to take AP Computer Science A.

Essential Questions

- How can I learn about technologies and computer science via the resources that are provided to be as open information on the Internet?
- Why open source? How does it affect me, the technology that I use, and the world as a whole from a social standpoint?
- What is more important, understanding the detailed syntax of a language, or how to use it to accomplish what I need to accomplish without learning every detail and esoteric piece of information about the language?

Outcomes

Upon successful completion of this course, students will:

- a. Understand several programming language concepts and technologies and their syntactical and conceptual differences.
- b. Design and implement programs using different programming environments.
- c. Use the Netbeans IDE for OOP exploration and Java program creation.
- d. Enter, edit, compile, run, test, and debug Java programs using Netbeans.
- e. Understand open source concepts and technologies
- f. Know and apply the basic syntax and grammar of Java for problem solving.
- g. Learn how to use the Java API for the purpose of obtaining information about Java classes.
- h. Apply the use of objects to the formulation of problem solutions.
- i. Design classes that model specific concepts, including mathematical concepts.
- j. Build database oriented software using NetBeans programming environment.
- k. Create complete Java programs by applying object oriented design techniques.

Instructional Strategies

The course will be taught with multiple concepts in mind. Lectures will always be accompanied with hands on TA/Fellows supported programming assignments to help students learn lectured concepts immediately. All of the tasks and projects that the students will work on will be team oriented. Most of the “testing” portion will involve the students in teams of 3-4 playing engaging games in order to ensure the understanding of each chapter’s topics. Some programming tasks will require the students to work in two or more groups. Peer programming is a popular concept in industry-based software engineering and that will be utilized in this class.

For major programming project, students will be broken up into teams so that they can learn how to break assignments into doable statements of work (sows). The projects will be organized into work that can be broken down into various different components (i.e. programming, writing, analysis, pseudo code/logic, algorithm development). This strategy will help each student to gain an understanding of all components of the project, while simultaneously focusing on a specific contribution to ensure the overall success of the project.

Resources and Materials

- P.J. Deitel, *Java How to Program: Early Objects Version*, Prentice Hall, 8th Edition. ISBN 0136053068
- Netbeans software development environment: <http://netbeans.org/>
- Robocode: <http://robocode.sourceforge.net/>
- MySQL database: www.mysql.org

Students should bring two notebooks, one spiral and one black and white composition book.

Student Assessment

- Pre-Assessment: *Details to follow in class*
- CTD Grading Scale:

A+	100-97%	A	96-93%	A-	92-90%
B+	89-87%	B	86-83%	B-	82-80%
C+	79-77%	C	76-73%	C-	72-70%
D+	69-67%	D	66-63%	D-	62-60%
F	below 60%				
- Breakdown of final grade: (15%), participation in classroom competitions (15%), Composition book review (15%), projects (30%), and programming assignments (30%)
- Post-Assessment: *Details to follow in class*

Schedule

Date(s)	Topic(s)	In-class Activities	Graded Assignments and/or Assessment
June 27, 2011 (Monday)	Introductions Chapter 1 – History of Programming and the Internet	Student and teacher introductions Syllabus review Handouts and other administrative tasks.	Pre – assessment test given.

		Slide based chapter review.	
June 28, 2011 (Tuesday)	Class broken up into teams. Chapter 1 review Introduction to Netbeans. Chapter 2 – Introduction to Java Programming	Review via class participation exercise. programming projects. Netbeans overview Chapter 2 review with slides	Programs due. Class participation
June 29, 2011 (Wednesday)	Computer Science project. Chapter 2 Review Chapter 3 - Intro to Classes and Objects	Slide based Lecture Self study and prep for the Comp Science Project	Class Participation
June 30, 2011 (Thursday)	Intro to Netbeans part II Chapter 3 review Chapter 4 /5 Control Statements pt 1 & 2	Slides based lecture Write and review simple programs	Coding Class participation
July 1, 2011 (Friday)	Review Chapters 4 and 5.	Individual programming Assignment	Coding
July 4, 2011 (Monday)	View Linux and Open Source videos, and prepare for discussion.		Continue Coding
July 5, 2011 (Tuesday)	Analysis of videos. Chapter 6- Methods a deeper look Introduction to Robocode programming assignments	Lecture Programming	Coding Class participation
July 6, 2011 (Wednesday)	Robocode assignments, review and tweaking. Chapter 6 review. Chapter 7 Lecture – Arrays and ArrayLists.	Lectures Code review	Coding Class participation
July 7, 2011 (Thursday)	Chapter 7 review. Chapter 8 – Classes and Objects – A deeper Look. Formalize teams for Robocode competitions	Coding	Coding
July 8, 2011 (Friday)	Chapter 8 Review. Robocode assignment documentation and	Coding/Lecture	Assignment and write-up work

	demonstration		
July 11, 2011 (Monday)	Chapter 14- GUI components Chapter 27 – Networking Chapter 28 Accessing Databases.	Lectures and testing exercises	Assignment work
July 12, 2011 (Tuesday)	Chapter 14,27,28 review Introduction of final project assignment. Pairing groups into two's.	Work on final project	Programming
July 13, 2011 (Wednesday)	Hands on work on final project continued	Work on final project	Programming
July 14, 2011 (Thursday)	Presentations of Final Projects. Course Review.	Final programming work	Programming
July 15, 2011 (Friday)	Post Assessment Test	Farewell and Performance Discussions	Good bye.

CTD Statement on Third-Party Web Sites

Instructors are required to thoroughly review any third-party web sites they intend to use in their courses for inappropriate content. However, because web content continuously changes, CTD disclaims any responsibility for any of the content contained on third-party web sites used in course materials. If you become aware of anything that may be inappropriate, please notify CTD staff immediately.