

BERKELEY HEIGHTS PUBLIC SCHOOLS
BERKELEY HEIGHTS, NEW JERSEY

**GOVERNOR LIVINGSTON HIGH SCHOOL
MATHEMATICS DEPARTMENT**

**AP COMPUTER SCIENCE-JAVA
#0352**

Curriculum Guide

September 2003

Dr. Richard G. Bozza, Superintendent
Mrs. Judith A. Rattner, Assistant Superintendent
Mr. Brian Heineman, District Supervisor

Developed by: Steven R. Ciarrocca

This curriculum may be modified through varying techniques,
strategies, and materials, as per an individual student's
Individualized Educational Plan (IEP).

Approved by the Berkeley Heights Board of Education
at the regular meeting held on _____.

TABLE OF CONTENTS

	Page
Philosophy and Rationale.....	1
Course Proficiencies.....	2
Course Objectives	2
Student Proficiencies	3
Methods of Evaluation.....	5
Course Outline/Student Objectives	6
Resources/Activities Guide.....	10
Suggested Audio Visual/Computer Aids.....	11
Suggested Materials	12
Resources for Students	12
Resources for Teacher	12

Addendum

New Jersey Core Curriculum Content Standards – Mathematics

New Jersey Core Curriculum Content Standards – Technological Literacy

**New Jersey Core Curriculum Content Standards – Career Education and
Consumer, Family, and Life
Skills**

*** New Jersey Core Curriculum Content Standards can also be viewed at
www.state.nj.us**

PHILOSOPHY/RATIONALE

Advanced Placement Computer Science-Java is a full year course. It is intended to serve both as an introductory course for computer science majors and as a course for students who will major in other disciplines that require significant involvement with computer programming. It is comparable to the introductory sequence in computer science offered by most college and university computer science departments.

The course will cover the Computer Science A/AB topics as outlined by the College Board. It will focus on a formal and in-depth study of data structures, algorithms, and object oriented programming.

The prerequisite for Advanced Placement Computer Science Java is Introduction to Java. Five credits are earned for successful completion of the course. The New Jersey Core Curriculum Content Standards for Mathematics, Technological Literacy, and Career Education and Consumer, Family, and Life Skills have been integrated throughout the curriculum.

COURSE PROFICIENCIES

COURSE OBJECTIVES

1. To provide students with knowledge and appreciation of the advanced concepts of object oriented programming.
2. To teach students the concepts of various data structures and demonstrate how each is implemented.
3. To give students an opportunity to design and implement solutions to practical and complex problems.
4. To develop the students' ability to determine the efficiency and validity of their code.
5. To provide students the experience of reading a large program and describing the design and development process leading to such a program. (An example of such a program is the Marine Biology Case Study developed by the College Board.)
6. To demonstrate the impact computer programming has on all aspects of life.
7. To encourage students to use computers and computer programming in an ethical manner.

STUDENT PROFICIENCIES

The student will be able to:

1. Apply various searching and sorting algorithms and describe the efficiency of each using Big-O notation.
(4.2/A4; 4.3/A3, D3; 8.1/B10,11; 9.1/B3)
2. Apply object oriented programming concepts such as inheritance, encapsulation, information hiding, and polymorphism to design programs.
(4.2/A4; 4.3/A3, C1, D3; 4.4/A2, B3; 4.5/A1-3,5, B1-4, C3,4, D2-4; 8.1/B2,10,11; 8.2/B3,5, 6; 9.1/B3,6; 9.2/A1,3)
3. Understand the concept of an abstract class and derive concrete classes from an abstract class.
(4.2/A4; 4.3/A3, C1, D3; 4.4/A2; 4.5/A1-3,5, B1-4, C3,4, D2-4; 8.1/B2,10,11; 8.2/B3,5,6; 9.1/B3,6; 9.2/A1,3)
4. Apply *ArrayList* and *LinkedList* libraries in an effective way and understand the advantages and disadvantages of each structure.
(4.2/A4; 4.3/A3, C1, D3; 4.4/A2, B3; 4.5/A1-3,5, B1-4, C3,4, D2-4; 8.1/B2,10,11; 8.2/B3,5, 6; 9.1/B3,6; 9.2/A1,3)
5. Use iterators to traverse a list.
(8.1/B10,11; 9.1/B3,6)
6. Understand and apply the concepts of stacks and queues and perform basic operations on each.
(4.2/A4; 4.3/A3, C1, D3; 4.4/A2, B3; 4.5/A1-3,5, B1-4, C3,4, D2-4; 8.1/B2,10,11; 8.2/B3,5, 6; 9.1/B3,6; 9.2/A1,3)
7. Recover from a runtime error by “throwing” an exception.
(8.1/B10,11; 8.2/B3,5,6; 9.1/B3; 9.2/A1)
8. Understand the concept of recursion, identify the essential elements of a recursive function, and apply recursion to situations in which recursion is most appropriate.
(4.2/A4; 4.3/A3, C1, D3; 4.5/A1-3,5, B1-4, C3,4, D2-4; 8.1/B2,10,11; 8.2/B3,5,6; 9.1/B3,6; 9.2/A1,3)
9. Explain the effects that a recursive function can have on the efficiency of a program in terms of both execution time and memory usage.
(9.1/B6; 9.2/A1)

STUDENT PROFICIENCIES (continued)

10. Understand the concepts of trees, identify various properties of a tree, explain various relationships between nodes of a tree, and traverse a tree in preorder, inorder, and postorder.
(4.5/C3; 9.2/A3)

11. Implement and apply trees using a variety of methods.
(4.2/A4; 4.3/A3, C1, D3; 4.4/A2, B3; 4.5/A1-3,5, B1-4, C3,4, D2-4; 8.1/B2,10,11; 8.2/B3,5, 6; 9.1/B3,6; 9.2/A1,3)

12. Implement and apply priority queues using a variety of methods.
(4.2/A4; 4.3/A3, C1, D3; 4.4/A2, B3; 4.5/A1-3,5, B1-4, C3,4, D2-4; 8.1/B2,10,11; 8.2/B3,5, 6; 9.1/B3,6; 9.2/A1,3)

13. Express the efficiency of algorithms using Big-O notation and recognize the importance of classifying algorithms in terms of their efficiency.
(4.2/A4; 4.3/D3; 4.5/A1-3,5, B1-4, C3,4, D2-4; 9.1/B6)

14. Use Java's Swing package to design programs that have a graphical user interface.
(8.1/B2,10,11; 8.2/B3,5,6; 9.1/B3,6; 9.2/A1,3)

METHODS OF EVALUATION

Methods of evaluation that are used in Advanced Placement Computer Science Java include but are not limited to:

1. Tests.
2. Quizzes.
3. Individual programs.
4. Group programs.
5. Written homework.
6. Midterm and final exams.

SCOPE AND SEQUENCE
COURSE OUTLINE/STUDENT OBJECTIVES

The student will be able to:

N.J. Core Curriculum Standards	Indicators	Course Outline/Student Objectives *(see note at end of outline)
4.2 4.3 8.1 9.1	A4 A3 D3 B10,11 B3	I. Searching and Sorting Algorithms A. Search an Array Using the Sequential Search Algorithm B. Search an Array Using the Binary Search Algorithm C. Sort an Array Using the Selection Sort Algorithm D. Sort an Array Using the Insertion Sort Algorithm E. Sort an Array Using the Mergesort Algorithm F. Sort an Array Using the Quicksort Algorithm G. Describe the Efficiency of all of the Sort and Search Algorithms Using Big-O Notation
4.2 4.3 4.4 4.5 8.1 8.2 9.1 9.2	A4 A3 C1 D3 A2 B3 A1,2,3,5 B1,2,3,4 C3,4 D2,3,4 B2,10,11 B3,5,6 B3,6 A1,3	II. Object Oriented Concepts and Design A. Design Program Involving Multiple, Interacting Class Objects B. Understand Class Hierarchy and Inheritance C. Define One Class (subclass) as an Extension of Another Class (base class) D. Know the Difference Between an Abstract Class and a Concrete Class E. Derive Concrete Classes from an Abstract Class F. Understand the Advantages of Encapsulation and Information Hiding G. Understand and Implement Polymorphism Techniques
4.2 4.3 4.4 4.5 8.1 8.2 9.1 9.2	A4 A3 C1 D3 A2 B3 A1,2,3,5 B1,2,3,4 C3,4 D2,3,4 B2,10,11 B3,5,6 B3,6 A1,3	III. Lists and Iterators A. Understand and Apply the Methods of the <i>List</i> Interface B. Use the <i>ArrayList</i> Library Class to Manage Data C. Use the <i>LinkedList</i> Library Class to Manage Data D. Design and Apply Classes That Implement Various Types of Linked Lists. 1. Singly Linked Lists 2. Doubly Linked Lists 3. Circularly Linked Lists E. List the Advantages and Disadvantages of Using Each Data Structure F. Use Iterators to Traverse a List

4.2	A4	<p>IV. Stacks and Queues</p> <p>A. Understand the Concept of a Stack</p> <p>B. Perform Push and Pop Operations on a Stack</p> <p>C. Design Various Implementations of a Stack and Apply Them to Practical Problems</p> <ol style="list-style-type: none"> 1. Array based 2. Linked List based <p>D. Understand the Concept of a Queue</p> <p>E. Perform Enqueue and Dequeue Operations on a Queue</p> <p>F. Design Various Implementations of a Queue and Apply Them to Practical Problems</p> <ol style="list-style-type: none"> 1. Array based 2. Linked List based <p>G. Recover from a Runtime Error by “Throwing” an Exception</p>	
4.3	A3		
	C1		
	D3		
4.4	A2		
	B3		
4.5	A1,2,3,5		
	B1,2,3,4		
	C3,4		
	D2,3,4		
8.1	B2,10,11	<p>V. Recursion</p> <p>A. Understand the Concept of Recursion</p> <p>B. Identify the Essential Elements of a Recursive Function</p> <ol style="list-style-type: none"> 1. Base case 2. Recursive call 3. Makes progress toward base case <p>C. Recognize Situations in Which a Recursive Function is Most Appropriate and Design Such a Function</p> <p>D. Explain the Effects That a Recursive Function Can Have on the Efficiency of a Program in Terms of Both Execution Time and Memory Usage</p>	
8.2	B3,5,6		
9.1	B3,6		
9.2	A1,3		
4.2	A4		<p>VI. Binary Trees</p> <p>A. Understand the Concepts of General Trees, Binary Trees, and Binary Search Trees</p> <p>B. Identify the Root and Leaves of a Tree and Determine its Height</p> <p>C. Identify the Parent, Children, Ancestors and Descendants of any Node in a Tree Structure</p> <p>D. Implement Various Traversals on a Tree Structure</p> <ol style="list-style-type: none"> 1. Preorder 2. Inorder 3. Postorder <p>E. Use a Tree to Store and Manage Data</p> <p>F. Explain the Link Between Tree Structures and Recursion</p> <p>G. Implement Binary Search Trees (BST) Using Both the <i>TreeSet</i> and <i>TreeMap</i> Library Classes</p> <p>H. Design and Apply Classes That Implement BST</p>
4.3	A3		
	C1		
	D3		
4.4	A2		
	B3		
4.5	A1,2,3,5		
	B1,2,3,4		
	C3,4		
	D2,3,4		
8.1	B2,10,11	<p>VII. Lookup Tables and Hashing</p> <p>A. Use a Lookup Table to Store and Retrieve Data</p> <p>B. Use a Hash Table to Store and Retrieve Data</p> <p>C. Determine an Appropriate Hash Function for a Set of Data</p>	
8.2	B3,5,6		
9.1	B3,6		
9.2	A1,3		

4.4	A2 B3	<p>VII. Lookup Tables and Hashing (continued)</p> <p>Values</p> <p>D. Utilize the <i>HashSet</i> and <i>HashMap</i> Library Classes to Hash a Set of Values</p> <p>E. Employ Techniques to Handle Collisions Within a Hash Table</p> <ol style="list-style-type: none"> 1. Linear Probing 2. Quadratic Probing 3. Chaining <p>F. Analyze the Effects That Collisions Have on the Performance of a Search in a Hash Table</p>
4.5	A1,2,3,5 B1,2,3,4 C3,4 D2,3,4	
8.1	B2,10,11	
8.2	B3,5,6	
9.1	B3,6	
9.2	A1,3	
4.2	A4	<p>VIII. Priority Queues</p> <p>A. Understand the Concept of a Priority Queue</p> <p>B. Discuss Several Different Implementations of a Priority Queue and the Advantages and Disadvantages of Each</p> <ol style="list-style-type: none"> 1. Array based 2. Linked List based <p>C. Understand the Concept of a Heap and Implement a Heap Using Arrays</p> <p>D. Use a Heap to Implement a Priority Queue</p> <p>E. Use a Heap Implemented Priority Queue to Implement the Heapsort</p>
4.3	A3 C1 D3	
4.4	A2 B3	
4.5	A1,2,3,5 B1,2,3,4 C3,4 D2,3,4	
8.1	B2,10,11	
8.2	B3,5,6	
9.1	B3,6	
9.2	A1,3	
4.2	A4	
4.3	D3	
4.5	A1,2,3,5 B1,2,3,4 C3,4 D2,3,4	
9.1	B6	
8.1	B2,10,11	
8.2	B3,5,6	<p>X. Graphical User Interface (GUI) Components and Events</p> <p>A. Use Components of Java's <i>Swing</i> Package to Design Programs That Have a Graphical User Interface</p> <ol style="list-style-type: none"> 1. <i>JButton</i> 2. <i>JCheckBox</i> 3. <i>JComboBox</i> 4. <i>JLabel</i>
9.1	B3,6	
9.2	A1,3	
9.2	A1,3	

		<p>X. Graphical User Interface (GUI) Components and Events (continued)</p> <ul style="list-style-type: none">5. <i>JPasswordField</i>6. <i>JRadioButton</i>7. <i>JSlider</i>8. <i>JTextArea</i>9. <i>JTextField</i>10. <i>JToggleButton</i> <p>B. Recognize the Connection Between GUI Programs and Event Driven Programming</p> <p>C. Use <i>actionListener</i> Objects to Capture and Process Events for <i>Swing</i> Components</p> <p>D. Place <i>Swing</i> Components on a Container Object Using a Variety of Layout Managers</p> <ul style="list-style-type: none">1. <i>FlowLayout</i>2. <i>GridLayout</i>3. <i>BorderLayout</i>4. <i>BoxLayout</i>
--	--	---

RESOURCES/ACTIVITIES GUIDE

Web sites:

www.apl.jhu.edu/~hall/java

Java programming resources collected by Marty Hall at Johns Hopkins University.

www.csc.calpoly.edu/~mliu/javaResources.html

Resources for teaching Java collected by Dr. Mei-Ling Liu at California Polytechnic State University.

www.cs.ukc.ac.uk/teaching/resources/java/

Java resources collected at CS Dept, the University of Kent at Canterbury.

<http://apcentral.collegeboard.com/>

The official College Board AP web site.

<http://cs.colgate.edu/APCSWeb/APCSJava.html>

Chris Nevison's "Unofficial APCS Web Page" for Java.

SUGGESTED AUDIO VISUAL/COMPUTER AIDS

1. Proxima LCD Projector.
2. Microsoft PowerPoint Software.
3. Microsoft Word Software.
4. Microsoft Internet Explorer Software.

SUGGESTED MATERIALS

Resources for Students

Text:

Litvin and Litvin
Java Methods AB: Data Structures
Skylight Publishing
2003

Supplemental Materials:

Litvin
Be Prepared for the AP Computer Science Exam in Java
Skylight Publishing
2003

Software:

JCreator Pro version 2.5
Xinox Software
JCreator Pro is an integrated development environment for Java.

Jcreator LE version 2.5
Xinox Software
JCreator LE is a freeware integrated development environment for Java available for download at jcreator.com

Website:

<http://www.skylit.com/javamethods/index.html>
The *Java Methods AB* companion web site contains both student and teacher resources.

Resources for Teacher

Text:

Litvin and Litvin
Java Methods AB: Data Structures
Skylight Publishing
2003

Supplemental Materials:

Litvin
Be Prepared for the AP Computer Science Exam in Java
Skylight Publishing
2003

SUGGESTED MATERIALS (continued)

Deitel and Deitel

Java: How to Program

Prentice Hall

2002

Software:

JCreator Pro version 2.5

Xinox Software

JCreator Pro is an integrated development environment for Java.

Web site:

<http://www.skylit.com/javamethods/index.html>

The *Java Methods AB* companion web site contains both student and teacher resources.