

BERKELEY HEIGHTS PUBLIC SCHOOLS
BERKELEY HEIGHTS, NEW JERSEY

GOVERNOR LIVINGSTON HIGH SCHOOL
MATHEMATICS DEPARTMENT

INTRODUCTION TO JAVA
#0351

Curriculum Guide

September 2002

Dr. Richard G. Bozza, Superintendent
Mrs. Judith A. Rattner, Assistant Superintendent
Miss Marian Szabo, 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 – Workplace Readiness	

PHILOSOPHY/RATIONALE

Introduction to Java (#0351) serves as a preliminary course in which students learn programming techniques using the Java programming language. It also serves as a prerequisite for Advanced Placement Computer Science, which is a college level course in programming and will be taught using Java beginning in the 2003-2004 school year.

The curriculum uses Java to teach students object oriented programming techniques. It also stresses a systematic approach to solving problems through the use of Java. The language features contained in the curriculum include user defined and built-in functions, variable declaration and usage, input/output objects, selection structures, repetition structures, and use and design of class objects.

The prerequisite for Introduction to Java is Computer Science 1. Two and a half credits are earned for successful completion of this one semester course. The New Jersey Core Curriculum Content Standards for Mathematics and the New Jersey Core Curriculum Content Standards for Workplace Readiness have been integrated throughout the curriculum.

COURSE PROFICIENCIES

COURSE OBJECTIVES

1. To provide students with knowledge and appreciation of the concepts of object oriented programming.
2. To teach students a systematic approach to solving problems that they can apply in other aspects of their lives.
3. To give students an opportunity to model real life objects and simulate their interaction using object oriented programming.
4. To develop students ability to convert their ideas into useful, working code.
5. To enable students to recognize the close relationship between mathematics and computer science.
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. Identify and describe the various hardware components of a computer system. 8.4 B/2; 8.5 A/1
2. Identify and describe operating systems and other software applications. 8.4 B/1,2
3. Understand the binary and hexadecimal number systems and be able to convert numbers from one system to another. 4.1 A/1, B/1
4. List and explain the essential components of a network. 8.4 B/2; 8.5 A/1
5. Describe and implement effective algorithms. 4.2 A/4; 4.3 C/1; 4.5 A/1-5, B/1-3, C/3,4, D/2-4, E/2; 8.1 B/2,3; 8.3 A/1; 8.4 D/1
6. Identify and describe the three types of programming languages and distinguish between compilers and interpreters. 8.4 B/2; 8.6 A/11
7. Understand the principles of object-oriented programming and explain the advantages of designing programs using an object-oriented model. 4.2 A/4; 4.3 C/1; 4.5 A/1-5, B/1-3, C/3,4, D/2-4, E/2; 8.1 B/2,3; 8.3 A/1; 8.4 D/1
8. Use appropriate rules of style to write code that is easy to understand and debug. 4.5 B/1,2; 8.1 B/3
9. Make effective use of variables and constants of primitive and class type. 4.1 A/1, B/1; 4.2 D/1,2; 4.3 C/1, D/1,3; 4.5 A/1-5, C/3,4, D/2-4, E/2, F/1; 8.3 A/1; 8.4 B/2, D/1; 8.5 A/3
10. Understand the concept of scope and determine the scope of a variable. 4.3 C/14.5 A/1-5, D/2-4, F/1; 8.3 A/1; 8.4 B/2, D/1; 8.5 A/3
11. Use input and output methods to process data. 4.5 B/1,2; 8.1 B/3; 8.4 B/2, D/1
12. Use *if* and *switch* structures in programs. 4.1 A/2, B/2; 4.2 A/4; 4.5 A/1-5, C/3,4, D/2-4, F/1; 8.3 A/1; 8.4 B/2, D/1; 8.5 A/3
13. Design and use classes given specifications about the nature of the class. 4.2 A/4 4.3 C/1; 4.5 A/1-5, B/1-3, C/3,4, D/2-4, E/2, F/1; 8.1 B/2,3; 8.3 A/1; 8.4 B/2, D/1, 8.5 A/3

STUDENT PROFICIENCIES (continued)

14. Determine what fields, methods, and constructors should be defined when designing a class and implement each in code. 4.2 A/4; 4.3 C/1; 4.5 A/1-5, B/1-3, C/3,4, D/2-4, E/2, F/1; 8.1 B/2,3; 8.3 A/1; 8.4 B/2, D/1; 8.5 A/3
15. Understand the distinction between a value parameter and a reference parameter and determine when each type is used. 4.2 A/4; 4.3 C/1; 4.5 A/1-5, D/2-4, 8.1 B/2; 8.3 A/1; 8.4 D/1
16. Use *string* and *stringbuffer* objects to input, process, and output strings of characters. 4.2 A/4; 4.5 A/1-5, F1; 8.3 A/1; 8.4 B/2, D/1; 8.5 A/3
17. Declare and manipulate one and two-dimensional arrays. 4.2 A/4; 4.3 C/1 4.5 A/1-5, C3,4, D2-4, E2, F1; 8.3 A/1; 8.4 B/2, D/1; 8.5 A/3
18. Use the *while*, *do..while*, and *for* structures to loop code. 4.2 A/4; 4.3 C/1; 4.5 A/1-5, C3,4, D2-4, E2, F1; 8.3 A/1; 8.4 B/2, D/1; 8.5 A/3

METHODS OF EVALUATION

Methods of evaluation that are used in Introduction to Java include but are not limited to:

1. Tests.
2. Quizzes.
3. Individual Projects.
4. Group Projects.
5. Written Homework.
6. Final Exam.

SCOPE AND SEQUENCE
COURSE OUTLINE/STUDENT OBJECTIVES

The student will be able to:

N.J. Core Curriculum Standards	Indicators	Course Outline/Student Objectives
4.1 8.4	A1 B1 B1,2	I. An Introduction to Hardware, Software, and the Internet A. explain what a CPU is and how it works B. understand the unit measurements of memory C. understand the difference between RAM and ROM memory D. identify various types of storage devices and why such devices are necessary E. identify various input/output devices F. identify various operating systems and explain how one organizes the resources of the computer for its users G. explain what applications are and identify various types of applications H. demonstrate how numbers and characters are represented in a computer's memory I. understand the binary and hexadecimal number systems and be able to convert numbers from one system to another J. list and explain the essential components of a network K. identify various Internet applications and describe what they do
4.2 4.3 4.5 8.1 8.3 8.4	A4 C1 A1-5 B1-3 C3,4 D2-4 E2 B2,3 A1 B2 D1	II. An Introduction to Software Development A. describe what an algorithm is and identify the properties of an effective algorithm B. express an algorithm using pseudocode C. identify and describe the three types of programming languages D. explain what a compiler is and how it translates source code into an executable file E. explain what an interpreter is and explain how it differs from a compiler F. know how and why Java uses a mixed compiler/interpreter approach to translate source

8.6	A11	<p>II. An Introduction to Software Development (continued)</p> <p>code</p> <p>G. understand the importance of developing reusable code</p> <p>H. recognize the principles of object oriented programming and explain the advantages of designing programs using an object oriented model</p>
4.5 8.1	B1,2 B3	<p>III. Java, Syntax, and Style</p> <p>A. use comments to explain sections of code</p> <p>B. recognize reserved words defined in Java</p> <p>C. choose appropriate names for classes, field methods, and variables that he/she defines in a program</p> <p>D. make effective use of white space and indentation in code</p>
4.1 4.2 4.3 4.5 8.3 8.4 8.5	A1 B1 D1,2 C1 D1,3 A1-5 C3,4 D2-4 E2 F1 A1 B2 D1 A3	<p>IV. Data Types, Variables, and Arithmetic</p> <p>A. declare variables of various types and assign them values</p> <p>B. list and describe the primitive data types</p> <p>C. make a distinction between variable objects and constant objects and demonstrate appropriate use of both</p> <p>D. understand the concept of scope and determine the scope of a variable</p> <p>E. perform arithmetic operations on variables with knowledge of the order in which the operations will be evaluated</p> <p>F. recognize how operations act differently when used with different data types</p> <p>G. use cast operators to convert values from one type to another</p> <p>H. use input methods to get information from the user</p> <p>I. use output methods to display results of program</p>
4.1 4.2 4.5	A2 B2 A4 A1-5 C3,4 D2-4	<p>V. Boolean Expressions and Selection Structures</p> <p>A. design <i>if</i> and <i>switch</i> structures for programs that require multiple possibilities to be handled</p> <p>B. model conditions using the relational operators (>, <, <=, !=, ==)</p> <p>C. design more complex conditions using the logical operators &&, , and !</p>

8.3 8.4 8.5	F1 A1 B2 D1 A3	V. Boolean Expressions and Selection Structures (cont.) D. recognize situations in which short circuit evaluation can simplify conditions
4.2 4.3 4.5 8.1 8.3 8.4 8.5	A4 C1 A1-5 B1-3 C3,4 D2-4 E2 F1 B3 A1 B2 D1 A3	VI. Classes, Objects, and Events A. understand the structure of a class of objects B. describe the relationship between derived classes and base classes C. use the <i>import</i> statement to use Java library classes and interfaces in programs D. identify, describe, and use fields, constructors and methods of a class E. design and use classes given specifications about the nature of the class F. distinguish between <i>public</i> and <i>private</i> designations G. understand what an event-driven program is and explain why Java is suitable for programming such applications
4.2 4.3 4.5 8.1 8.3 8.4 8.5	A4 C1 A1-5 B1-3 C3,4 D2-4 E2 F1 B3 A1 B2 D1 A3	VII. Methods, Constructors, and Fields A. determine what fields, methods, and constructors should be defined when designing a class B. design and call methods both with and without parameters C. overload methods of a class when appropriate D. design and call constructors, including copy constructors E. overload constructors of a class when appropriate F. use the <i>new</i> statement to initialize an object G. understand the difference between static and non-static fields and methods, and determine when each designation is appropriate H. understand the distinction between a value parameter and a reference parameter, and determine when each type is used I. use the <i>return</i> statement to quit a method and, if necessary, return a value to the calling method
4.2 4.5	A4 A1-5 F1	VIII. Strings A. understand the type of data <i>string</i> and <i>stringbuffer</i> objects contain B. explain the difference between <i>string</i> and

8.3 8.4 8.5	A1 B2 D1 A3	<p>VIII. String (continued)</p> <p><i>stringbuffer</i> objects</p> <p>C. use the <i>length</i>, <i>charAt</i>, <i>substring</i>, <i>indexOf</i>, <i>lastIndexOf</i>, <i>trim</i>, <i>replace</i>, <i>toUpperCase</i>, <i>toLowerCase</i>, and concatenation methods to manipulate <i>string</i> objects</p> <p>D. use the <i>length</i>, <i>charAt</i>, <i>setCharAt</i>, <i>capacity</i>, <i>append</i>, <i>insert</i>, <i>substring</i>, <i>indexOf</i>, <i>replace</i>, <i>toUpperCase</i>, <i>toLowerCase</i>, <i>delete</i>, <i>deleteCharAt</i>, and concatenation methods to manipulate <i>stringbuffer</i> objects</p> <p>E. use the compare methods to compare <i>string</i> and <i>stringbuffer</i> objects</p> <p>F. convert <i>string</i> and <i>stringbuffer</i> objects to numbers and vice versa</p> <p>G. use character methods to manipulate <i>string</i> and <i>stringbuffer</i> objects</p>
4.2 4.3 4.5 8.3 8.4 8.5	A4 C1 A1-5 C3,4 D2-4 E2 F1 A1 B2 D1 A3	<p>IX. Arrays</p> <p>A. understand what an array object is and how it is stored in memory</p> <p>B. declare and create one and two dimensional arrays</p> <p>C. access and assign an array's elements using subscripts</p> <p>D. access the length of an array</p> <p>E. pass array arguments to methods</p> <p>F. use a loop to traverse all of the elements of an array</p>
4.2 4.3 4.5 8.3 8.4 8.5	A4 C1 A1-5 C3,4 D2-4 E2 F1 A1 B2 D1 A3	<p>X. Iterative Statements</p> <p>A. use the <i>while</i>, <i>do..while</i>, and <i>for</i> structures to loop code</p> <p>B. design both sentinel and counter controlled looping structures</p> <p>C. apply nested looping structures effectively</p> <p>D. make proper use of the <i>break</i> statement</p>

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: An Introduction to Object Oriented Programming
Skylight Publishing
2001

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

Resources for Teacher

Text:

Litvin and Litvin
Java Methods: An Introduction to Object Oriented Programming
Skylight Publishing
2001

Software:

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

Web site:

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