COURSE TITLE

Introduction to Java

LENGTH

One Semester
Grades 10-12

DEPARTMENT

Computer Department
Barbara O'Donnell, Supervisor

SCHOOL

Rutherford High School

DATE

Fall 2016

INTRODUCTION TO JAVA

I. Introduction/Overview/Philosophy

The Java language is the standard teaching programming language in most universities today. This course is an introduction to object-oriented programming in Java. It focuses on problem solving and algorithm development in a graphical environment. The key elements of a Java program, objects, classes, and methods are explored and modified in prewritten Java applications and employed to the design and implementation of user-defined classes. The students then use the Greenfoot environment to help further develop their Java abilities.

The prerequisites for this course are the successful completion of: Computer Programming I and Computer Programming C++.

II. Objectives

Course Outline:

I. Computer Systems
- A. Describe relationship between hardware and software
- B. Identify basic computer hardware and what it does
- C. Explain how the hardware components execute programs and manage data
- D. Describe how computers are connected together into networks to share information
- E. Explain the importance and responsibility of the World Wide Web

II. Writing Your First Programs
- A. Introduce the Java programming language
- B. Describe the steps involved in program compilation and execution
- C. Vocabulary of Java
  1. Reserved words
  2. Identifiers
  3. Syntax and semantics
- D. Data types
- E. Formatting output

III. Objects and Primitive Data
- A. Define the difference between primitive data and objects
- B. Declare and use variables
- C. Perform mathematical computations
- D. Create objects and use them
- E. Use class libraries and import packages

IV. Introduction to Graphics
- A. Explore the difference between a Java application and a Java applet
- B. Modify graphic classes
- C. Create graphical programs that draw shapes
- D. Execute applets using the Web

V. Program Statements
- A. Discuss program development steps
- B. Define the flow of control through a program
- C. Learn to use if/nested if statements
- D. Define expressions that allow for more complex decisions
- E. Increment/decrement and assignment operators

VI. Iteration
- A. Learn to use loops
  1. The for loop
  2. The while loop
- B. Trace the execution of a nested loop
- C. How to avoid infinite loops
- D. Using the StringTokenizer class in loops

VII. Writing Classes
- A. Define classes
- B. Identify the Anatomy of a Class
  1. The return statement
  2. Parameters
  3. Constructors
  4. Local data
- C. Method overloading and method decomposition
- D. Object relationships
  1. Association
  2. Association between objects of the same class
  3. Aggregation

VII. Arrays
- A. Format of Arrays
  1. Array indexing
  2. Declaring and using arrays
  3. Automatic bounds checking
  4. Off-by-one-error
  5. Initializer lists
  6. Array parameters
- B. Searching
  1. Binary search
  2. Linear search
  3. Sequential search
- C. Sorting
  1. Bubble sort
  2. Insertion sort

VIII. Greenfoot
- A. Interacting with Greenfoot
  1. The Greenfoot interface
  2. Creating a World with classes and objects
  3. Making objects act

4. Running a scenario
5. Invoking methods

B. Movement and Key Control
1. Using Move, turn methods
2. Using isKeyDown

C. Detecting and Removing Actors, and Making Methods
1. Using getOneObjectOffset, remove methods
2. Using Refactoring behaviors

D. Saving the World, Making and Playing Sound
1. Save the World method
2. Recording and Playing Sounds

E. Adding a Randomly Moving Enemy
1. Using getRandomNumber
2. Using getHeight, getWidth

F. Designing Scenarios

Student Outcomes:

Upon completion of the course, students will demonstrate the ability to:

1. describe the relationship between hardware and software
2. define various types of software and how they are used
3. identify basic computer hardware and explain what it does
4. describe how the hardware components execute programs and manage data
5. describe how computers are connected together into networks to share information
6. explain the importance of the Internet and the World Wide Web
7. introduce the Java programming language
8. describe the steps involved in program compilation and execution
9. introduce graphics and their representation
10. define the difference between primitive data and objects
11. declare and use variables
12. perform mathematical operations
13. create objects and use them
14. explore the difference between a Java application and a Java applet
15. create graphical programs that draw shapes
16. discuss basic program development steps
17. define the flow of control through a program
18. learn to use if statements
19. define expressions that let us make complex decisions
20. learn to use while and for statements
21. use conditionals and loops to draw graphics
22. define classes that act like blueprints for new objects, made of variables and methods
23. explain encapsulation and Java modifier
24. explore the details of method declarations
25. review method invocation and parameter passing
26. explain and use method overloading
27. learn to divide complicated methods into simpler, supporting methods
28. describe relationships between objects

29. define and use arrays.
30. describe how arrays and array elements are passed as parameters
31. explore how arrays and other objects can be combined to manage complex information.
32. explore searching and sorting with arrays.
33. use Greenfoot environment to implement objects, invoke methods, and create scenarios of games and simulations.

New Jersey Student Learning Standards

TECHNOLOGY

Standard 8.1: Educational Technology: All students will use digital tools to access, manage, evaluate, and synthesize information in order to solve problems individually and collaboratively and to create and communicate knowledge.

Standard 8.2: Technology Education, Engineering, and Design: All students will develop an understanding of the nature and impact of technology, engineering, technological design, and the designed world, as they relate to the individual, global society, and the environment.

> Strand E. Computational Thinking: Programming: Computational thinking builds and enhances problem solving, allowing students to move beyond using knowledge to creating knowledge.

21ST CENTURY LIFE AND CAREERS

Standard 9.2: Career Awareness, Exploration, and Preparation

Standard 9.3 – Career & Technical Education (CTE)
Pathway: Programming & Software Development (IT-PRG)

III. Proficiency Levels

This course is open to grades 10-12.

IV. Methods of Assessment

Student Assessment

The teacher will provide a variety of assessments during the course of the year. Among these are: homework, laboratory exercises, teacher-made tests and quizzes, and long-term projects.

Curriculum/Teacher Assessment

The teacher will provide the subject area supervisor with suggestions for changes on an ongoing basis.

V. Grouping

The prerequisite for Introduction to Java is successful completion of Computer Programming C++.

VI. Articulation/Scope & Sequence/Time Frame

Course length is one semester and is offered to students in grades 10-12.

VII. Resources

Texts/Supplemental Reading/References

Cocking, Cara, Lewis, John, and Loftus, William, Java Software Solutions, Boston, MA: Addison Wesley, 2004.

Horstmann, Cay Java Concepts, AP Edition, Hoboken, NJ: John Wiley & Sons, Inc., 2005.

Cocking, Cara Lewis, John, and Loftus, William, Java Software Solutions for AP Computer Science A, 2nd Edition, Boston, MA: Addison Wesley, 2006.

Introduction to Programming in Greenfoot, Kolling, Pearson, 2009

www.codingBat.org

www.greenfoot.org

VIII. Methodologies

Much of the class time is spent in lab work and on programming problems to be completed. When group instruction is necessary, topics are taught using the computer projection system in conjunction with student classwork.

IX. Suggested Activities

- Laboratory programming problems
- Class presentations
- Cooperative programming projects

X. Interdisciplinary Connections

Connections are made to mathematics by using a variety of arithmetic formulas, as well as higher mathematical concepts. Connections are also made to the disciplines of art, English, and science by means of incorporation of these ideas into programming projects.

XI. Differentiating Instruction for Students with Special Needs: Students with Disabilities, English Language Learners, and Gifted & Talented Students

Differentiating instruction is a flexible process that includes the planning and design of instruction, how that instruction is delivered, and how student progress is measured. Teachers recognize that students can learn in multiple ways as they celebrate students' prior knowledge. By providing appropriately challenging learning, teachers can maximize success for all students.

Examples of Strategies and Practices that Support:

Students with Disabilities

- Use of visual and multi-sensory formats
- Use of assisted technology
- Use of prompts
- Modification of content and student products
- Testing accommodations
- Authentic assessments

Gifted & Talented Students

- Adjusting the pace of lessons
- Curriculum compacting
- Inquiry-based instruction
- Independent study
- Higher-order thinking skills
- Interest-based content
- Student-driven
- Real-world problems and scenarios

English Language Learners

- Pre-teaching of vocabulary and concepts
- Visual learning, including graphic organizers
- Use of cognates to increase comprehension
- Teacher modeling
- Pairing students with beginning English language skills with students who have more advanced English language skills
- Scaffolding
    - •word walls
    - •sentence frames
    - •think-pair-share
    - •cooperative learning groups

XII. Professional Development

The teacher will continue to improve expertise through participation in a variety of professional development opportunities.

XIII. Curriculum Map

Class	September/February	October/March	November/April	December/May	January/June
Intro to Java	Sec (2.0-2.6) Intro to Objects Using Objects/Print method/abstraction String Literals/Concatenation Escape sequence Variables and Assignment/Constants Primitive Data Types/ Integers and floating points/Booleans/ Characters Arithmetic Expressions/ Operator precedence Data conversion Creating Objects/String class/Wrapper classes Listings (2.1-2.8) PreLab/Lab Exercises (2.1-2.6)	Sec (2.7-2.8) Class Libraries and packages/Import declaration/ Random class Invoking Class Methods/ Math class/Scanner Formatting Output/ Number Format /Decimal Format Listings (2.1-2.8) PreLab/Lab Exercises (2.7-2.9) Sec (3.0-3.4) Program Development Control Flow If Statement/ Equality/Relational Operators/If-else/Block statements/nested if Boolean Expressions Revisited/ Comparing characters and strings/comparing float point values Listings (3.1-3.4) PreLab/Lab Exercises (3.2-3.4)	Sec (3.4-3.9) More Operators/ Increment/ Decrement The While Statement Infinite loops/Nested loops/ String Tokenizer class For Statement/ comparing loops Program Development Revisited Listings (3.5-3.13) PreLab/Lab Exercises (3.4-3.8) Sec (4.0-4.4) Objects Revisited/Classes/ Anatomy of a Class/Instance data/ Encapsulation/ Visibility Modifiers Anatomy of a Method/ Return/ Parameters/ Constructors/Local data Method Overloading Method Decomposition Listings (4.1-4.9) PreLab/Lab Exercises: Coin Class/Bank Account/Tracking Grades	Sec (4.5) Object Relationships/ Association/ Aggregation Listings (4.10-4.14) PreLab/Lab Exercises: Representing Names Arrays(indexing)(6.0) Arrays of Objects(6.1) Searching(6.2) Sorting(6.3) Interacting with Greenfoot The Greenfoot interface Creating a World with classes and objects Making objects act Running a scenario Invoking methods	Movement and Key Control Using Move , turn methods Using isKeyDown Detecting and Removing Actors, and Making Methods Using getOneObjectOffset, remove methods Using Refactoring behaviors Saving the World, Making and Playing Sound Save the World method Recording and Playing Sounds Adding a Randomly Moving Enemy Using getRandomNumber Using getHeight, getWidth Designing Scenarios Review for Final Exam