COURSE TITLE

Introduction to Java

LENGTH

One Semester
Grades 10-12

DEPARTMENT

Computer Department
Barbara O'Donnell, Supervisor

SCHOOL

Rutherford High School

DATE

September 10, 2018

Initial BOE Approval Date (Born on): 5/11/2015

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:

- describe the relationship between hardware and software
- define various types of software and how they are used
- identify basic computer hardware and explain what it does
- describe how the hardware components execute programs and manage data
- describe how computers are connected together into networks to share information
- explain the importance of the Internet and the World Wide Web
- introduce the Java programming language
- describe the steps involved in program compilation and execution
- introduce graphics and their representation
- define the difference between primitive data and objects
- declare and use variables
- perform mathematical operations
- create objects and use them
- explore the difference between a Java application and a Java applet
- create graphical programs that draw shapes
- discuss basic program development steps
- define the flow of control through a program
- learn to use if statements
- define expressions that let us make complex decisions
- learn to use while and for statements
- use conditionals and loops to draw graphics
- define classes that act like blueprints for new objects, made of variables and methods
- explain encapsulation and Java modifier
- explore the details of method declarations
- review method invocation and parameter passing
- explain and use method overloading
- learn to divide complicated methods into simpler, supporting methods
- describe relationships between objects
- define and use arrays.
- describe how arrays and array elements are passed as parameters
- explore how arrays and other objects can be combined to manage complex information.
- explore searching and sorting with arrays.
- use Greenfoot environment to implement objects, invoke methods, and create scenarios of games
- and simulations.

New Jersey Student Learning Standards

CAREER READY PRACTICES

CRP1 Act as a responsible and contributing citizen and employee

Career-ready individuals understand the obligations and responsibilities of being a member of a community, and they demonstrate this understanding every day through their interactions with others. They are conscientious of the impacts of their decisions on others and the environment around them. They think about the near-term and long-term consequences of their actions and seek to act in ways that contribute to the betterment of their teams, families, community and workplace. They are reliable and consistent in going beyond the minimum expectation and in participating in activities that serve the greater good.

CRP2 Apply appropriate academic and technical skills

Career-ready individuals readily access and use the knowledge and skills acquired through experience and education to be more productive. They make connections between abstract concepts with real-world applications, and they make correct insights about when it is appropriate to apply the use of an academic skill in a workplace situation.

CRP4 Communicate clearly and effectively and with reason.

Career-ready individuals communicate thoughts, ideas, and action plans with clarity, whether using written, verbal, and/or visual methods. They communicate in the workplace with clarity and purpose to make maximum use of their own and others' time. They are excellent writers; they master conventions, word choice, and organization, and use effective tone and presentation skills to articulate ideas. They are skilled at interacting with others; they are active listeners and speak clearly and with purpose. Career-ready individuals think about the audience for their communication and prepare accordingly to ensure the desired outcome.

CRP6 Demonstrate creativity and innovation

Career-ready individuals regularly think of ideas that solve problems in new and different ways, and they contribute those ideas in a useful and productive manner to improve their organization. They can consider unconventional ideas and suggestions as solutions to issues, tasks or problems, and they discern which ideas and suggestions will add greatest value. They seek new methods, practices, and ideas from a variety of sources and seek to apply those ideas to their own workplace. They take action on their ideas and understand how to bring innovation to an organization.

CRP8 Utilize critical thinking to make sense of problems and persevere in solving them

Career-ready individuals readily recognize problems in the workplace, understand the nature of the problem, and devise effective plans to solve the problem. They are aware of problems when they occur and take action quickly to address the problem; they thoughtfully investigate the root cause of the problem prior to introducing solutions. They carefully consider the options to solve the problem. Once a solution is agreed upon, they follow through to ensure the problem is solved, whether through their own actions or the actions of others.

CRP11 Use technology to enhance productivity

Career-ready individuals find and maximize the productive value of existing and new technology to accomplish workplace tasks and solve workplace problems. They are flexible and adaptive in acquiring new technology. They are proficient with ubiquitous technology applications. They understand the inherent risks-personal and organizational-of technology applications, and they take actions to prevent or mitigate these risks.

TECHNOLOGY STANDARDS

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 COLLABORATE AND TO CREATE AND COMMUNICATE KNOWLEDGE.
8.1.12.B.2 - Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
8.1.12.D.5 - Analyze the capabilities and limitations of current and emerging technology resources and assess their potential to address personal, social, lifelong learning, and career needs.
8.1.12.F.1 - Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.

STANDARD 8.2: TECHNOLOGY EDUCATION, ENGINEERING, DESIGN, AND COMPUTATIONAL THINKING – PROGRAMMING: ALL STUDENTS WILL DEVELOP AN UNDERSTANDING OF THE NATURE AND IMPACT OF TECHNOLOGY, ENGINEERING, TECHNOLOGICAL DESIGN, COMPUTATIONAL THINKING, AND THE DESIGNED WORLD AS THEY RELATE TO THE INDIVIDUAL, GLOBAL SOCIETY, AND THE ENVIRONMENT.
8.2.12.E.3 - Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
8.2.12.E.4 - Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).

STANDARD 9.3: CAREER AND TECHNICAL EDUCATION
PATHWAY: PROGRAMMING & SOFTWARE DEVELOPMENT (IT-PRG)
9.3.IT-PRG.6 Program a computer application using the appropriate programming language.

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. Suggested Activities

- Laboratory programming problems
- Class presentations
- Cooperative programming projects

IX. 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.

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, Students at Risk, 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.

Differentiating in this course includes but is not limited to:

Differentiation for Support (ELL, Special Education, Students at Risk)

- activity choice
- appeal to diverse learning styles
- choice to work with others or alone
- hands-on activities
- multimodal activities
- advance organizers
- pre-teaching vocabulary

- visual demonstrations, illustrations, and models
- work with checklists
- online video review
- peer teaching and support
- study guides
- partnering
- guided notetaking
- reteach and review
- demonstrations on smartboard
- peer mentoring on lab

Differentiation for Enrichment

- more complex tasks and problems
- higher expectations in assessment questioning
- independent extensions based on student interest, curiosity, and choice
- extended research and readings
- curriculum compacting
- flexible grouping on challenging exercises
- higher level questioning techniques
- facilitating classmates
- higher expectation for writing programs
- display and explain version of coded task
- peer mentoring

XII. Professional Development

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

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Computer Systems • Relationship between hardware and software • Execution of programs and management of data • Networks • Digital Citizenship	1 week	*For Support:* • activity choice • work with checklists • choice to work with others or alone *For Enhancement:* • independent extensions based on student interest, curiosity, and choice • extended research and readings higher level questioning techniques	*Standards:* CRP1, CRP4, 8.1.12.D.5, 8.1.12.F.1	*Formative Assessment:* • Research types of languages (machine - OOP) • Networking topics • Use of digital citizenship *Summative Assessment:* • Quiz (Presentation of chosen topic through powerpoint, visual display, essay, etc)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
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	6 weeks	*For Support:* - pre-teaching vocabulary - guided notetaking - visual demonstrations, illustrations, and models - hands-on activities *For Enhancement:* - higher level questioning techniques - higher expectation for writing programs more complex tasks and problems	*Standards:* CRP2, CRP6, 8.1.12.D.5, 8.2.12.E.3, 9.3.IT-PRG.6	*Formative Assessment:* - Questions of the Day using Google Classroom - based on current or previously learned content - Worksheets (Math Class - Worksheets (Class and Scope) (4) - PreLab Exercises (2.1-2.6) - Worksheet (Variables) (1) *Summative Assessment:* - Lab (Strings) - Quiz (Strings) - Test (Strings) - Quiz (Class / Scope) - Quiz (Variables) - Test (Syntax / Output) - Lab (OOP) - Lab (Math class) - Test (Class / Scope / Math class)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Class Libraries and Packages • Import declaration • Random class • Invoking Class Methods • Math class • Scanner • Formatting Output • Number Format • Decimal Format	3 weeks	*For Support:* • study guides • powerpoint reference • guided notetaking • reteach and review *For Enhancement:* • more complex tasks and problems • facilitating classmates • display and explain version of coded task	*Standards:* CRP6, CRP8, 8.1.12.F.1, 8.2.12.E.4	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheets (Random class) (2) • Worksheets (String methods) (2) *Summative Assessment:* • Lab (Math / Random Class) • String Lab#2 • Test (Math / Random Class) • Benchmark #1 on all material covered • Quiz (Strings) • Lab (Formatting Output)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
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	2 weeks	*For Support:* • appeal to diverse learning styles • peer teaching and support • hands-on activities *For Enhancement:* • more complex tasks and problems • higher expectation for writing programs • peer mentoring	*Standards:* CRP6, CRP8, 8.2.12.E.3, 8.2.12.E.4	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheets (If/Else/Nested Ifs) (3) • Listings (3.1-3.4) • PreLab Exercises (3.2-3.4) *Summative Assessment:* • Lab (If/Else/Nested Ifs) • Test (If/Else/Nested Ifs)
More Operators • Increment • Decrement • The While Statement • Infinite loops • Nested loops/ • String Tokenizer class • For Statement Comparing loops	1 week	*For Support:* • reteach and review • multimodal activities • visual demonstrations, illustrations, and models *For Enhancement:* • independent extensions based on student interest, curiosity, and choice • display and explain version of coded task • facilitating classmates	*Standards:* CRP2, CRP4, CRP6, 9.3.IT-PRG.6	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Program Development Revisited Listings (3.5-3.13) • PreLab Exercises (3.4-3.8) *Summative Assessment:* • Lab (Loops)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
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 • Association • Aggregation	2 weeks	*For Support:* • advance organizers • pre-teach vocabulary • demonstrations on smartboard *For Enhancement:* • extended research and readings • flexible grouping on challenging exercises • display and explain version of coded task	*Standards:* CRP1, CRP6, CRP11, 8.1.12.F.1, 8.2.12.E.3, 9.3.IT-PRG.6	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Listings (4.1-4.9) PreLab Exercises • Listings (4.10-4.14) PreLab Exercises *Summative Assessment:* • Lab Exercises Coin Class/Bank Account/Tracking Grades • Lab Exercises: Representing Names • Test (Objects, Classes, Methods)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Arrays • Indexing • Arrays of Objects • Searching • Sorting	2 weeks	*For Support:* • pre-teaching vocabulary • hands-on activities • online video review *For Enhancement:* • higher level questioning techniques • peer mentoring • higher expectations in assessment questioning	*Standards:* CRP6, CRP8, 8.1.12.B.2, 8.2.12.E.3, 9.3.IT-PRG.6	*Formative Assessment:* • Questions of the Day using Google Classroom - based on current or previously learned content • Worksheets (Arrays) (1) • CodingBat (Arrays exercises) (6) *Summative Assessment:* • MiniQuiz (Arrays) • Lab (Arrays1) • Lab (Arrays2) • Test (Arrays)

Unit Topic	Time Allocated	Differentiating Instruction for Students with Disabilities, Students at Risk of School Failure, English Language Learners, & Gifted & Talented Students	Standards	Assessments
Interacting with Greenfoot • The Greenfoot interface o Creating a World with classes and objects o Making objects act o Running a scenario o Invoking methods • Movement and Key Control o Using Move , turn methods o Using isKeyDown • Detecting and Removing Actors, and Making Methods o Using getOneObjectOffset, remove methods o Using Refactoring behaviors • Saving the World, Making and Playing Sound o Save the World method o Recording and Playing Sounds • Adding a Randomly Moving Enemy o Using getRandomNumber o Using getHeight, getWidth • Designing Scenarios	3 weeks	*For Support:* • activity choice • appeal to diverse learning styles • choice to work with others or alone *For Enhancement:* • curriculum compacting • independent extensions based on student interest, curiosity, and choice • facilitating classmates	*Standards:* CRP1, CRP2, CRP6, CRP8, 8.1.12.B.2	*Formative Assessment:* • Daily Progression of Greenfoot Scenario (Trick-the-Turtle) • Daily Progression of Greenfoot Scenario (Ketchup) • Daily Progression of Greenfoot Scenario (Pengu) *Summative Assessment:* • Lab (Trick-the-Turtle) • Test (Greenfoot - Objects, Classes, Methods) • Project (Game Simulation) • Benchmark #2 on material covered