Glen Ridge Public Schools – Computer Science Curriculum

**Course Title:** Computer Science 2 (CS2)

**Subject:** Computer Science

**Grade Level:** 9 - 12

**Duration:** 0.5 year (approx. 18 weeks)

**Prerequisite:** None

**Elective or Required:** Elective

**Computer Science, Engineering, and Technology Mission Statement**

Since computational thinking and problem solving are integral parts of our lives and 21st century learning, students must be actively involved in their Computer Science, Engineering, and Technology (CSET) education. The CSET curricula will emphasize thinking skills through a balance of computation, intuition, common sense, logic, design, analysis, and technology. Students will use a combination of technology and critical thinking to solve real-world problems. To achieve these goals, students will be taught a standards-based curriculum that is aligned with the New Jersey Curriculum Standards.

**Course Description:**

Computer Science 2 (CS2) is an elective course that expands students’ knowledge of the Python programming language and introduces students to Object Oriented Programming (OOP) using the Java programming language. Students in this course will have completed Computer Science 1 or an equivalent course and have an understanding of the software development process.

Greenfoot and BlueJ are used to teach Java and OOP. Greenfoot is a simple and easy to use graphical and interactive environment developed for beginner Java programmers. During the last unit of the course, students will be introduced to the BlueJ development environment. BlueJ is the main editor used in AP Computer Science A. CS2 is a pre-requisite course for Advanced Placement Computer Science A.

**Author:** Mayra Bachrach

**Date Submitted:** Summer 2016
Resources


Digital Resources

- How to think like a Computer Scientist: [http://interactivepython.org/runestone/static/thinkcspy/index.html](http://interactivepython.org/runestone/static/thinkcspy/index.html)
- Greenfoot Youtube channel: [http://www.youtube.com/user/18km?ob=5#g/u](http://www.youtube.com/user/18km?ob=5#g/u)
- Michael Kolling’s “Joy of Code” Greenfoot blog hosted by the University of Kent at: [http://blogs.kent.ac.uk/mik/category/joy-of-code/](http://blogs.kent.ac.uk/mik/category/joy-of-code/)
- Greenfoot site at: [http://www.greenfoot.org/door](http://www.greenfoot.org/door)
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises

Software Tools

- Python
- IDLE Python editor
- Java
- Greenfoot
- BlueJ
- Internet access
- Google Chrome (or other web browser, such as Safari, Internet Explorer, etc)
- Google Education products
## Curriculum Standards

### Technology

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1.12.B.2</td>
<td>Apply previous content knowledge by creating and piloting a digital learning game or tutorial.</td>
</tr>
<tr>
<td>8.2.12.E.1</td>
<td>Demonstrate an understanding of the problem-solving capacity of computers in our world.</td>
</tr>
<tr>
<td>8.2.12.E.2</td>
<td>Analyze the relationships between internal and external computer components.</td>
</tr>
<tr>
<td>8.2.12.E.3</td>
<td>Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).</td>
</tr>
<tr>
<td>8.2.12.E.4</td>
<td>Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).</td>
</tr>
</tbody>
</table>

### 21st Century Life and Careers

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.3.IT-PRG.4</td>
<td>Demonstrate the effective use of software development tools to develop software applications.</td>
</tr>
<tr>
<td>9.3.IT-PRG.5</td>
<td>Apply an appropriate software development process to design a software application.</td>
</tr>
<tr>
<td>9.3.IT-PRG.6</td>
<td>Program a computer application using the appropriate programming language.</td>
</tr>
<tr>
<td>9.3.IT-PRG.7</td>
<td>Demonstrate software testing procedures to ensure quality products.</td>
</tr>
<tr>
<td>9.3.IT-PRG.8</td>
<td>Perform quality assurance tasks as part of the software development cycle.</td>
</tr>
</tbody>
</table>

### 21st Century Life and Careers – Career Ready Practices

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRP1</td>
<td>Act as a responsible and contributing citizen and employee.</td>
</tr>
<tr>
<td>CRP2</td>
<td>Apply appropriate academic and technical skills.</td>
</tr>
<tr>
<td>CRP4</td>
<td>Communicate clearly and effectively and with reason.</td>
</tr>
<tr>
<td>CRP6</td>
<td>Demonstrate creativity and innovation.</td>
</tr>
<tr>
<td>CRP8</td>
<td>Utilize critical thinking to make sense of problems and persevere in solving them.</td>
</tr>
<tr>
<td>CRP12</td>
<td>Work productively in teams while using cultural global competence.</td>
</tr>
</tbody>
</table>

### Mathematics

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math Practices.1</td>
<td>Make sense of problems and persevere in solving them.</td>
</tr>
<tr>
<td>Math Practices.2</td>
<td>Reason abstractly and quantitatively.</td>
</tr>
<tr>
<td>Math Practices.3</td>
<td>Construct viable arguments and critique the reasoning of others.</td>
</tr>
<tr>
<td>Math Practices.4</td>
<td>Model with mathematics.</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Math Practices.5</td>
<td>Use appropriate tools strategically.</td>
</tr>
<tr>
<td>Math Practices.6</td>
<td>Attend to precision.</td>
</tr>
<tr>
<td>Math Practices.7</td>
<td>Look for and make use of structure.</td>
</tr>
<tr>
<td>Math Practices.8</td>
<td>Look for and express regularity in repeated reasoning.</td>
</tr>
</tbody>
</table>

**English Language Arts**

<table>
<thead>
<tr>
<th>RI.11-12.4</th>
<th>Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings; analyze how an author uses and refines the meaning of a key term or terms over the course of a text (e.g., how Madison defines faction in Federalist No. 10).</th>
</tr>
</thead>
<tbody>
<tr>
<td>RI.11-12.7</td>
<td>Integrate and evaluate multiple sources of information presented in different media or formats (e.g., visually, quantitatively) as well as in words in order to address a question or solve a problem.</td>
</tr>
<tr>
<td>RST.11-12.3</td>
<td>Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.</td>
</tr>
<tr>
<td>RST.11-12.4</td>
<td>Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 11-12 texts and topics.</td>
</tr>
<tr>
<td>RST.11-12.10</td>
<td>By the end of grade 12, read and comprehend science/technical texts in the grades 11-CCR text complexity band independently and proficiently.</td>
</tr>
<tr>
<td>NJSLSA.W4</td>
<td>Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.</td>
</tr>
<tr>
<td>NJSLSA.W9</td>
<td>Draw evidence from literary or informational texts to support analysis, reflection, and research.</td>
</tr>
</tbody>
</table>
Unit 1: Python Programming

Approximate # Of Weeks: 3 weeks

Essential Questions:

- How is a Python program opened, saved, modified and executed in the IDLE development environment?
- What is an iteration statement (loop)?
- What is a while loop?
- What is a for loop?
- How do loops help simplify program development?
- What is a list data structure?
- What are the common list operations (traverse, insert, replace, delete, etc.)?
- What is a function?
- How are functions defined in Python?
- How are values passed to a function?
- How does a function return a value?
- What is a Python standard library module?
- How are modules in the Python standard library used?
- Time Permitting:
  - What is a dictionary data structure?
  - How are values in a dictionary stored and retrieved?

Upon completion of this unit students will be able to:

- Explain and use the IDLE development environment.
- Explain and use while loops
- Explain and use for loops
- Explain and call functions
- Create functions that accept parameter values
- Create functions that return values
- Pass a parameter to a function
- Explain and use modules in the Python standard library.
- Explain, create and use a list.
- Explain and use the common operations on lists (insert, replace, delete, etc.).
- Time Permitting:
  - Explain and use a Dictionary to represent a matrix structure (such as a game board).
  - Explain and access the elements of a Dictionary.
  - Explain and derive a solution to a problem or challenge using the features of the Python programming language.
Interdisciplinary Standards


Activities:

- Lecture and follow along coding: Introduction to IDLE and review of the basic features of the Python language.
- Online tutorial Simple Python Data and Selection (How to think like a Computer Scientist):
- Labs from Simple Python Data Exercises (How to think like a Computer Scientist):
  - All Work and No Play
  - Compound Interest
  - Miles Per Gallon
  - Celsius to Fahrenheit
- Labs: IsEven and LeapYear from Selection Exercises (How to think like a Computer Scientist)
- Labs: Matinee Movie Tickets and Pie Eating Contest (Introduction to Computer Science using Java)
- Lecture and follow along coding: Loops
- Labs: Print the numbers from 1 to n using a loop, CountDown
- Lecture and follow along coding: Modules in the Python standard library
- Lab: Calculate the area of a circle
- Labs from Modules (How to think like a Computer Scientist):
  - Using a for loop print 10 random numbers
  - Using a for loop print random numbers between 25 and 35
- Project Guessing game with loops and random numbers
- Lecture and follow along coding: Functions
- Online tutorial Function (How to think like a Computer Scientist):
- Labs from Functions (How to think like a Computer Scientist):
  - Draw Polygon
  - Spiral
  - Area of a Circle function
  - Sumto(n)
- Lecture and follow along coding: Lists
- Online tutorial Lists (How to think like a Computer Scientist):
- Labs from Functions (How to think like a Computer Scientist):
  - Average of List of random numbers
  - Max of List
  - Sum of Squares
  - Number of odd numbers in a list
  - Number of even numbers in a list
  - How many words in list with length of 5
- Time permitting:
- Lecture and follow along coding: Dictionaries
- Online tutorial *Dictionaries (How to think like a Computer Scientist)*:
  - Labs from *Dictionaries (How to think like a Computer Scientist)*:
    - Number of Times each letter occurs
    - English to Pirates Translation
- Review Video lessons from *Automate the Boring Stuff with Python*:
  - Python basics in the IDLE interactive shell: https://www.youtube.com/watch?v=7qHMXu99d88
  - Saving and running a program in IDLE: https://www.youtube.com/watch?v=buMTH61Cngk
  - Boolean values and comparison operators: https://www.youtube.com/watch?v=4XA9CKJJbr4&feature=youtu.be
  - If, elif and else: https://www.youtube.com/watch?v=lWeCgEbk-Ro&feature=youtu.be
  - While loops: https://www.youtube.com/watch?v=885qKiiKisI&feature=youtu.be
  - For loop: https://www.youtube.com/watch?v=HFQGxh1jY3g&feature=youtu.be
  - Importing Modules: https://www.youtube.com/watch?v=xJLj6fWfw6k&feature=youtu.be
  - Functions: https://www.youtube.com/watch?v=WB4hJJkhLU&feature=youtu.be
  - Lists: https://www.youtube.com/watch?v=5n6o1MaXDoE&feature=youtu.be
  - For loops with lists: https://www.youtube.com/watch?v=umTnlPbYww&feature=youtu.be
  - Dictionaries: https://www.youtube.com/watch?v=o3m8XhnVDWs&feature=youtu.be
- Project: Rock-Paper-Scissors
- Project: Design, program and test a text base game in Python.

**STEAM Activities:**

- Lecture and follow along coding: Turtle Graphics
- Online Tutorial *Turtle Graphics (How to think like a Computer Scientist)*
- Labs from *Turtle Graphics (How to think like a Computer Scientist)*:
  - We like Python! 100 times
- Draw Triangle, Square, Hexagon and Octagon
- Draw Star
- Project: Design and program an original picture using Turtle Graphics

**Enrichment Activities:**

- Labs from *Simple Python Data Exercises (How to think like a Computer Scientist)*:
  - 24 Hour clock
  - Day of the week
- Project: Hangman
- Lab from *Functions (How to think like a Computer Scientist)*: Spirral

**Methods of Assessments/Evaluation:**

- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

**Resources:**

- How to think like a Computer Scientist:  
  [http://interactivepython.org/runestone/static/thinkcspy/index.html](http://interactivepython.org/runestone/static/thinkcspy/index.html)
- Sweigart, Al, *Automate the Boring Stuff with Python*,  
  [https://automatetheboringstuff.com/](https://automatetheboringstuff.com/)
- Kjell, Bradley, Central Connecticut State University *Introduction to Computer Science using Java*, Java 1.5 version, January 2006,  
  [http://chortle.ccsu.edu/cs151/cs151java.html](http://chortle.ccsu.edu/cs151/cs151java.html)
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
Unit 2: Structure and Syntax of Java and Using Greenfoot

Approximate # Of Weeks: 1 week

Essential Questions:

- What is Greenfoot and how does it relate to the Java programming language?
- What are the main sections of the Greenfoot development environment and how are they used?
- How is a Greenfoot project opened, saved, modified and executed?
- What is the Greenfoot API?
- What is a class?
- What is an object?
- What is a Greenfoot Actor?
- What is a Greenfoot World?
- What are the geometric properties of the Greenfoot world?
- What is the basic structure of a Java class?
- What are the basic syntax rules of the Java language?
- What is a method?
- What is a parameter?
- What methods of the Actor class are used to make Actors move and turn?

Upon completion of this unit students will be able to:

- Explain and use the main components of the Greenfoot programming environment.
- Explain and use the Greenfoot class.
- Explain and use the built-in methods of a Greenfoot Actor.
- Explain and use the built-in methods of a Greenfoot World.
- Explain and use the geometric properties of the Greenfoot world.
- Open, modify and execute applications using Greenfoot.
- Explain the difference between a class and an object.
- Explain and use the structure of a Java class.
- Explain and use the basic syntax rules of the Java language.
- Explain and avoid some of the common syntax errors made by Java programmers.
- Explain and call methods.
- Explain and pass parameters to methods
- Explain and use the Greenfoot API.

Interdisciplinary Standards

- 8.2.12.E.1, 8.2.12.E.2, 8.2.12.E.3, 8.2.12.E.4, Math Practices 1-8, 21st Century Career Practices 1, 2, 4, 6, 8 and 12, 9.3.IT.PRG.4,
Activities (All STEAM Activities):

- Lecture, class discussion and follow along coding
- Labs:
  - Wombat world exploration
  - Asteroids world exploration
  - HedgeHog world exploration.
  - Little crab – move and turn
- Project: Lady Bug
- Journal entry: Students will compare and contrast the difference between a class and an object
- Review video lessons from the Joy of Code (Greenfoot YouTube channel):
  - Classes and objects: [https://www.youtube.com/watch?v=tZNnYeqm5BU](https://www.youtube.com/watch?v=tZNnYeqm5BU)
  - Moving and turning: [https://www.youtube.com/watch?v=C3BIPoIG4Ok](https://www.youtube.com/watch?v=C3BIPoIG4Ok)

Enhancement Activities:

- Activity: Students will visit the Java API website and compare the Java API with the Greenfoot API.
- Activity: Students will visit the Greenfoot gallery and analyze the source code for completed projects.

Methods of Assessments/Evaluation:

- Unit quizzes.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

Resources:

- Greenfoot Youtube channel: [http://www.youtube.com/user/18km?ob=5#g/u](http://www.youtube.com/user/18km?ob=5#g/u)
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
Unit 3: Interacting Objects and Creating Methods

Approximate # Of Weeks: 3 weeks

Essential Questions:

- What are the primitive data types?
- How is a value passed to a method in Greenfoot?
- What is the return type of a method and why is it important?
- How does a Greenfoot method report a value?
- How does Greenfoot use inheritance?
- What is an inheritance hierarchy?
- What is a subclass?
- What methods of the Actor class are used to make Actors move, turn and detect the edge of the world?
- How is an “if” statement programmed in Greenfoot?
- How are comments coded in Java?
- What is dot notation?
- What is a static method and how is it called?
- How is a random number generated in Greenfoot?
- What are the comparison operators in Java?
- How is a new Greenfoot world created?
- How are new Actor classes created?
- What is a Greenfoot image?
- What is a String?
- How are new methods defined in a Java class?
- What are the Java naming conventions?
- How is a String represented in Java?
- How does a Greenfoot program interact with the Keyboard?
- How is sound added to a Greenfoot world?
- What Greenfoot method is used to end a program?

Upon completion of this unit students will be able to:

- Explain and use the primitive data types: int, double and boolean.
- Explain and call methods.
- Identify return types and parameters given a method signature.
- Explain and pass parameters to methods.
- Explain and use methods that return a value.
- Explain and identify inheritance relationships between Actors.
- Explain and use “if” statements.
- Explain and use comments in Java.
- Explain and use dot notation in calling methods.
- Explain and use the static methods of the Greenfoot class.
- Compare static and non-static methods.
- Explain and use random numbers.
• Explain and use the comparison operators.
• Explain and create new subclasses of the World class.
• Explain and create new subclasses of the Actor class.
• Explain and use Greenfoot images.
• Explain and define new methods in a Java class.
• Explain and use the java naming conventions.
• Explain and use Strings.
• Explain and use the isKeyDown Greenfoot method to detect key presses.
• Explain and use the playSound Greenfoot method to add sound.
• Explain and use the stop method to end a class.

**Interdisciplinary Standards**


**Activities (All STEAM Activities):**

• Lecture, class discussion and follow along coding
• Labs:
  o Little crab – move random
  o Little crab – turn left or right
  o Creating new subclasses
  o Add a lobster
  o Writing new methods
  o Keyboard control
  o Stop method
  o Adding sound
• Project: Trick the Turtle
• Project: Lady Bug and Spider
• Project: Friendly Cat
• Journal entry: Students will compare and contrast the difference between a class and an object
• Journal entry: Students will identify the main parts of a Java class.
• Review video lessons from the Joy of Code (Greenfoot YouTube channel):
  o Random behavior: [https://www.youtube.com/watch?v=-xKH34gMxZ4](https://www.youtube.com/watch?v=-xKH34gMxZ4)
  o Creating subclasses, methods, Greenfoot API: [https://www.youtube.com/watch?v=lev8xhaYUBU](https://www.youtube.com/watch?v=lev8xhaYUBU)
  o Sound: [https://www.youtube.com/watch?v=1NDWuosxEuk](https://www.youtube.com/watch?v=1NDWuosxEuk)
  o Structure of a class: [https://www.youtube.com/watch?v=21Btgjv9b1Q](https://www.youtube.com/watch?v=21Btgjv9b1Q)
**Enrichment Activities:**

- Trick the Turtle Enhancement
- Enhance the Little crab following Greenfoot tutorial (https://www.youtube.com/watch?v=lev8xhaYUBU)

**Methods of Assessments/Evaluation:**

- Unit tests.
- Unit quizzes.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

**Resources:**

- Greenfoot Youtube channel: http://www.youtube.com/user/18km?ob=5#g/u
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
Unit 4: Constructors, Instance variables and Creating objects

Approximate # Of Weeks: 3 weeks

Essential Questions:

- What is a constructor?
- What is the function of the constructor and when does it run?
- How are objects created using programming statements?
- How are Actors added to a world using programming statement?
- What is the GreenfootImage class?
- What is an instance variable?
- How is an assignment statement coded in Java?
- How are “if” statements and methods of the GreenfootImage class used to animate actors?
- What are the arithmetic operators in Java?
- How are instance variables used as counters?
- How is a Greenfoot application executed outside of the Greenfoot environment?

Upon completion of this unit students will be able to:

- Explain and use constructors.
- Explain and use the “new” keyword to call the constructor of a class.
- Explain and use the addObject method of a Greenfoot world to add new actors.
- Explain and use the GreenfootImage class.
- Explain and use instance variables.
- Explain and use assignment statements to initialize variables.
- Identify and use the arithmetic operators in Java.
- Explain and use instance variables as counters.
- Explain and export a Greenfoot world as a standalone Java application.

Interdisciplinary Standards


Activities (All STEAM Activities):

- Lecture, class discussion and follow along coding
- Labs:
  - Running a project outside of Greenfoot
  - Creating an object and adding objects to the world
• Editing images in Greenfoot
• Recording sound in Greenfoot
• A First look at Variables
• Object Interaction – Create a Bug
• Object Interaction – Eat a Bug
• Object Interaction – Add a Bug to the World
• Animating images
• Making a new project (Bouncing Balls)
• Default constructor (Bouncing Balls)
• Constructor with input (Bouncing Balls)
• Constructor and shift click (Bouncing Balls)
• Adding an Instance Variable (Bouncing Balls)

- Project: Falling Leaves
- Project: Boy Walking
- Project: Fat Cat walking
- Journal entry: Students will write a journal entry explaining the algorithm used to detect that two objects are touching
- Review video lessons from the Joy of Code (Greenfoot YouTube channel):
  - Variables: https://www.youtube.com/watch?v=apFNNMVK6u4
  - Counter variable: https://www.youtube.com/watch?v=WIifij-961o
  - Object Interaction: https://www.youtube.com/watch?v=F9BhAqeHzI
  - Bouncing Balls: https://www.youtube.com/watch?v=brBjVL4Mx2Q

**Enrichment Activities:**
- Fat Cat – Walk left and Right
- Fat Cat – Sleep
- Project: Follow the Joy of Code tutorial to add a graphical counter to the world: https://www.youtube.com/watch?v=WIifij-961o

**Methods of Assessments/Evaluation:**

- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

**Resources:**

- Greenfoot Youtube channel: http://www.youtube.com/user/18km?ob=5#g/u
• Greenfoot API at: 
• Teacher Webpage and YouTube channel
• Google Classroom/Drive class notes and exercises
Unit 5: Make Your Own World

Approximate # Of Weeks: 1 week

Essential Questions:

- What are the challenges encountered in developing computer applications?
- How are classes, objects, constructors, methods, variables and decision statements combined to develop a computer application?
- What are some of the issues to consider when designing applications involving user interactions?
- How can applications be tested to minimize or eliminate errors?
- How can applications be documented to help the user and other programmers understand and use the application?

Upon completion of this unit students will be able to:

- Explain and use the application development process.
- Apply programming knowledge to design and develop an original application.
- Explain issues associated with user interface design.
- Explain and use effective testing practices.
- Develop documentation for a more complex application.

Interdisciplinary Standards


Activities (All STEAM Activities):

- Working in pairs, students will submit a proposal for an original Greenfoot world.
- Working in pairs, students will design, develop and test an original Greenfoot world.
- Working in pairs, students will write specifications to document their original world.
- Students will share their projects with the class.

Methods of Assessments/Evaluation:

- Assessing students’ daily progress and participation.
• Grading completed proposal, project and presentation as per teacher rubric.

Resources:
• Greenfoot Youtube channel: http://www.youtube.com/user/18km?ob=5#g/u
• Michael Kolling’s “Joy of Code” Greenfoot blog hosted by the University of Kent at: http://blogs.kent.ac.uk/mik/category/joy-of-code/
• Greenfoot site at: http://www.greenfoot.org/door
• Greenfoot API at: http://www.greenfoot.org/files/javadoc/greenfoot/Greenfoot.html
  Teacher Webpage and YouTube channel
• Google Classroom/Drive class notes and exercises
Unit 6: Intro to Loops and Arrays

Approximate # Of Weeks: 2 weeks

Essential Questions:

- How are the logical operators, “and” “or”, represented in Java?
- What is a loop?
- What is a while loop?
- What is a for loop?
- What is a local variable?
- What is the scope of a variable?
- What is an Array?
- How are the individual elements of an Array referenced?
- What are some of the common errors programmers make when using Arrays?

Upon completion of this unit students will be able to:

- Explain and use the logical operators, && and ||.
- Explain and use while loops.
- Explain and use for loops.
- Compare and contrast local, parameter and instance variables.
- Identify and use the scope of a variable.
- Compare and contrast the scopes of local, parameter and instance variables.
- Explain and use an Array.
- Explain and use the index and elements of an Array.
- Explain and use programming statements to create, traverse and insert elements into an Array.
- Explain and avoid the common errors that programmers make when using Arrays.

Interdisciplinary Standards


Activities (All STEAM Activities):

- Lecture, class discussion and follow along coding
- Labs:
  - Introduction to while loops
  - Using While loops (Bee)
Using While loops 2 (Bee and Ladybug)
- Intro to For loops
- Using For loops
- Piano Keys labs 1 through 3
- Intro to Arrays labs 1 through 3 (Fruits)
- Project: Animals (using Array of images)
- Journal entry: Students will write a journal entry explaining an Array and describing the benefits of using an Array.

Enrichment Activities:

- Project: Piano Enhancement (Black Keys)

Methods of Assessments/Evaluation:

- Unit quizzes.
- Unit test.
- Programming labs
- Projects
- Responses to discussion questions
- Verbal Assessment
- Think/Pair/Share
- Thumbs Up/Thumbs Down
- Exit slips

Resources:

- Greenfoot Youtube channel: http://www.youtube.com/user/18km?ob=5#g/u
- Teacher Webpage and YouTube channel
- Google Classroom/Drive class notes and exercises
Unit 7: Console input and output in BlueJ

Approximate # Of Weeks: 2 weeks

Essential Questions:

- What are the main components of the BlueJ development environment and how are they used?
- How is a project opened, modified and saved in BlueJ?
- How is Java class written, compiled and executed in BlueJ?
- What is console output?
- How is Java source code translated to an executable form?
- What is byte code?
- What is the starting point of a Java program?
- How is text output printed to the console?
- How is the Scanner class used to interact with the user?

Upon completion of this unit students will be able to:

- Explain and use the BlueJ development environment
- Explain, write and compile a Java class in BlueJ.
- Explain and use the main method.
- Explain and use the Java API.
- Explain and use the steps in the Java compilation process.
- Explain and use the Scanner and System classes for input and output.
- Given a problem statement, develop a solution using the Java language using the BlueJ development environment.

Interdisciplinary Standards


Activities:

- Lecture, demo of BlueJ, class discussion and follow along coding
- BlueJ Labs (all on Google classroom):
  - Using BlueJ
  - Creating methods and passing parameters with BlueJ
  - Creating Arrays without Initializer List with BlueJ
  - Using BlueJ Arrays with Input
  - Practice with Loops and Arrays in BlueJ
  - Practice with Loops and Arrays labs part 2 and 3
- Project: Text Based version of the Guessing game
• Journal entry: Students will write a journal entry explaining the comparing and contrasting writing Java programs in Greenfoot and BlueJ.

**Enrichment Activities:**

• Project: Text based version of Rock Paper Scissors

**Methods of Assessments/Evaluation:**

• Unit quizzes.
• Unit test.
• Programming labs
• Responses to discussion questions
• Verbal Assessment
• Think/Pair/Share
• Thumbs Up/Thumbs Down
• Exit slips

**Resources:**

• Teacher Webpage and YouTube channel
• Google Classroom/Drive class notes and exercises
Unit 8: Final Project

Approximate # Of Weeks: 3 weeks

Essential Questions:

- What are the challenges encountered in developing computer applications?
- How can all the concepts and techniques learned in the course be applied to developing a complex game application?
- What are the issues to consider when designing applications involving user interactions?
- How can applications be tested to minimize or eliminate errors?
- How can applications be documented to help the user and other programmers understand and use the application?

Upon completion of this unit students will be able to:

- Explain and use the application development process.
- Apply programming concepts and techniques to design and develop a complex game application.
- Explain issues associated with user interface design.
- Explain and use effective testing practices.
- Develop documentation for a complex application.

Interdisciplinary Standards


Activities (All STEAM Activities):

- Working in pairs, students will submit a proposal for an original game application.
- Working in pairs, students will design, develop and test an original game application.
- Working in pairs, students will write specifications to document their original application.
- Students will share their projects with the class.

Methods of Assessments/Evaluation:

- Assessing students’ daily progress and participation.
• Grading completed proposal, project and presentation as per teacher rubric.

Resources:
• Greenfoot Youtube channel:
  http://www.youtube.com/user/18km?ob=5#g/u
• Michael Kolling’s “Joy of Code” Greenfoot blog hosted by the University of Kent at: http://blogs.kent.ac.uk/mik/category/joy-of-code/
• Greenfoot site at: http://www.greenfoot.org/door
• Greenfoot API at: http://www.greenfoot.org/files/javadoc/greenfoot/Greenfoot.html
• Teacher Webpage and YouTube channel
• Google Classroom/Drive class notes and exercises