GEORGIA | PEACH STATE PATHWAYS

Career, Technical, & Agricultural Education

BUSINESS & COMPUTER SCIENCE

PATHWAY: Computing

COURSE: Beginning Programming

UNIT: 3.3 Classes and Objects

INTRODUCTION

Annotation:

Grade(s):

X	9th
X	10th
X	11th
X	12th

Time: 5 weeks

Author: Jason Naile

Additional Author(s):

Students with Disabilities:

For students with disabilities, the instructor should refer to the student's IEP to be sure that the accommodations specified are being provided. Instructors should also familiarize themselves with the provisions of Behavior Intervention Plans that may be part of a student's IEP. Frequent consultation with a student's special education instructor will be beneficial in providing appropriate differentiation.

FOCUS STANDARDS

GPS Focus Standards:

BCS-BP-5 Students will design algorithms and programming solutions for a variety of computational problems.
a. Apply, test, analyze, and adjust problem-solving algorithms.
b. Design, test, analyze, and adjust coding solutions based on problem-solving algorithms.
c. Analyze and discuss coding solutions for elements of thoroughness and correctness.
d. Describe how recursion can be used to solve a problem.

BCS-BP-6 Students will design solutions for simple programs using basic programming techniques and constructs.
a. Implement techniques such as conditional statements, iterative statements, and variables to solve simple problems.
b. Utilize basic mathematical expressions to solve simple problems.
c. Use arrays and lists where appropriate.
d. Comment programs to aid program readability.
e. Test and debug simple programs.
f. Analyze and explain simple programs involving fundamental programming constructs.

BCS-BP-7 Students will use and develop algorithms to solve simple problems.
a. Develop algorithms to solve simple problems using psuedocode and/or flowcharts.
b. Interpret algorithms expressed in psuedocode and/or flowcharts to code solutions to simple problems.
c. Discuss the importance of algorithms in problem solving.
d. Identify properties of well-written algorithms in solving problems.
e. Interpret algorithms to write code that will implement searching and sorting techniques.

BCS-BP-8 Students will demonstrate knowledge of the relationships between classes.
a. Utilize Class, Responsibilities, and Collaborator (CRC) in problem analysis.
b. Create UML diagrams to illustrate relationships between classes.
c. Describe and explain the implementation of "is-a" and "has-a" relationships.
d. Describe how using classes implements the ideas of encapsulation and information hiding.

BCS-BP-9 Students will write programs that process one-dimensional arrays.
a. Calculate the average of the contents of an array.
b. Manipulate each element of an array.
c. Manipulate a section of an array.

BCS-BP-10 Students will write programs that process two-dimensional arrays.
a. Manipulate each element of a two-dimensional array.
b. Manipulate a section of a two-dimensional array.
c. Explain the difference between row major and column major two-dimensional arrays.
d. Explain how to process an array in a row-first or column-first manner.

BCS-BP-11 Students will write programs that process lists.
a. Manipulate each element of a list.
b. Add items to a list.

BCS-BP-12 Students will demonstrate knowledge of the basics of stacks and queues.
a. Explain the purpose of a call stack.

b. Describe the purpose of a printer queue.
c. Compare and contrast stacks and queues.

GPS Academic Standards:

ELA11W1 The student produces writing that establishes an appropriate organizational structure, sets a context and engages the reader, maintains a coherent focus throughout, and signals a satisfying closure.

ELA11W2 The student demonstrates competence in a variety of genres.

MM3A7 Students will understand and apply matrix representations of vertex-edge graphs.

ELA11LSV1 The student participates in student-to-teacher, student-to-student, and group verbal interactions.

SCSh3 Students will identify and investigate problems scientifically.

MM3P1 Students will solve problems using appropriate technology.

National Standards:

UNDERSTANDINGS & GOALS

Enduring Understandings:

- **At the conclusion of this unit, students should understand the different data structures that can be used in Computer Programming. Not only should students know the different data structures, but they should also be able to determine when to use the appropriate data structures. Finally, students should be able to implement the various data structures.**

Essential Questions:

- What is an array?
- How do one-dimensional arrays differ from two-dimensional arrays?
- How do you create, populate and traverse arrays?
- What are row major and column major two dimensional arrays?
- What can you do with a list?
- What are stacks and queues?
- What is the purpose of a call stack and a printer queue?

Knowledge from this Unit:

- Students will be able to explain the difference between a one-dimensional and two-dimensional array.
- Students will be able to create, populate, and traverse both one-dimensional and two-dimensional arrays (performing various calculations on each).
- Students will be able to compare and contrast row major and column major two-dimensional arrays.

- Students will be able create, add items to, and use various methods of a list.
- Students will be able to describe the differences between stacks and queues.
- Students will be to explain the purpose of a call stack and printer queue.

Skills from this Unit:

- Students will work collaboratively to accomplish a task.
- Students will display proper interpersonal communication techniques.
- Students will conduct self-evaluation.
- Students will design algorithms.
- Students will implement basic programming constructs when appropriate.
- Students will test and debug programs.

ASSESSMENT(S)

Assessment Method Type: Select one or more of the following. Please consider the type(s) of differentiated instruction you will be using in the classroom.

____ Pre-test
X Objective assessment - multiple-choice, true- false, etc.
_x_ Quizzes/Tests
__ Unit test
____ Group project
____ Individual project
____ Self-assessment - May include practice quizzes, games, simulations, checklists, etc.
__ Self-check rubrics
__ Self-check during writing/planning process
__ Journal reflections on concepts, personal experiences and impact on one's life
__ Reflect on evaluations of work from teachers, business partners, and competition judges
__ Academic prompts
__ Practice quizzes/tests
X Subjective assessment/Informal observations
__ Essay tests
__ Observe students working with partners
_x_ Observe students role playing
____ Peer-assessment
__ Peer editing & commentary of products/projects/presentations using rubrics
__ Peer editing and/or critiquing
____ Dialogue and Discussion
__ Student/teacher conferences
__ Partner and small group discussions
__ Whole group discussions
__ Interaction with/feedback from community members/speakers and business partners
____ Constructed Responses
__ Chart good reading/writing/listening/speaking habits
__ Application of skills to real-life situations/scenarios
____ Post-test

Assessment(s) Title: **Written tests, self evaluation, and informal checks**

Assessment(s) Description/Directions:
Throughout the unit, teachers should use assessments at their discretion. Appropriate assessments throughout the unit include conducting informal observation and asking students to evaluate themselves. Additionally, teachers should quiz and test students as they deem appropriate.

Attachments for Assessment(s):
Web Resources: http://www.bluepelicanjava.com/

Web Resource Title: Blue Pelican Java

Web Resource Description: A free e-book offering programming practices, projects, quizzes, and tests is available for teachers covering data structures.

LEARNING EXPERIENCES

Instructional planning: Include lessons, activities and other learning experiences in this section with a brief description of the activities to ensure student acquisition of the knowledge and skills addressed in the standards. Complete the sequence of instruction for each lesson/task in the unit.

Sequence of Instruction

1. **Identify the Standards. Standards should be posted in the classroom for each lesson.**
2. **Review Essential Questions.**
    - What is an array?
    - How do one-dimensional arrays differ from two-dimensional arrays?
    - How do you create, populate and traverse arrays?
    - What are row major and column major two dimensional arrays?
    - What can you do with a list?
    - What are stacks and queues?
    - What is the purpose of a call stack and a printer queue?
3. **Identify and review the unit vocabulary.**
4. **Assessment Activity.**

Sequence of Instruction and Learning:

(Based on a 50 minute period)
Week 1: Introduction to Data Structures and Introduction to One Dimensional arrays (allow for practice time)
Week 2: Two Dimensional arrays (allow for practice time)
Week 3: Lists---discuss associated methods (allow for practice time)
Week 4: Stacks and Queues (allow practice time)
Week 5: Tic-Tac-Toe Program
Week 6 (if needed): Tic-Tac-Toe Program

Attachments for Learning Experiences: Please list.

Notes & Reflections:

CULMINATING PERFORMANCE TASK (Optional)

Culminating Unit Performance Task Title: **Tic-Tac-Toe Program**

Culminating Unit Performance Task Description/Directions/Differentiated Instruction: Students will create a Tic-Tac-Toe program that uses a two-dimensional array. Students may work individually or in pairs to complete this project. Full specification and a grading checklist are attached.

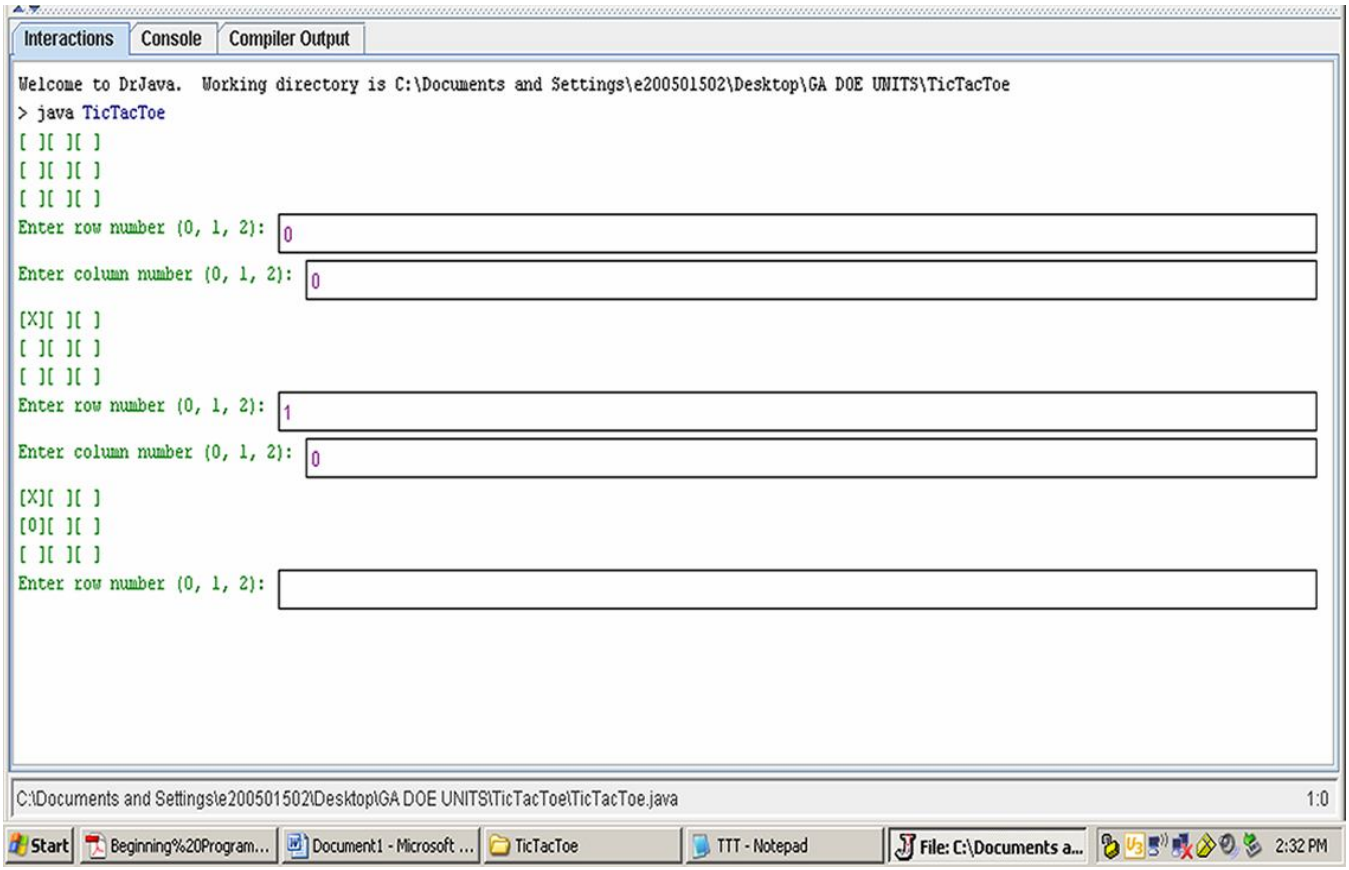
Attachments for Culminating Performance Task:

Rubric for Performance Task:

Programming Tic-Tac-Toe

Directions: Using a Two-Dimensional array program a text based Tic-Tac-Toe game in Java. The program should include a constructor for the Tic-Tac-Toe game, a play method, a display board method, a make move method and a winner method. One class should be named Tic-Tac-Toe and the other should be a tester/driver.

The player will play the game by inputting the row and column they wish to make their corresponding mark. A screen shot is below.



Grading Checklist

Task	Possible Points	Points Earned
Documentation/Comments	10	
Proper Style/Conventions used	10	
Program include all constructors and methods listed above	20	
Program is functional	20	
Program uses 2-D array to store moves	20	
Nested loop is used to check for winner	20	

UNIT RESOURCES

Web Resources:

Attachment(s):

Materials & Equipment:

Computer

Network storage space

Internet access

Java Software Development Kit

Java Integrated Development Environment (Dr. Java, Eclipse, BlueJ)

Desktop Publishing Software

Index cards

What 21st Century Technology was used in this unit:

	Slide Show Software		Graphing Software		Audio File(s)
	Interactive Whiteboard		Calculator		Graphic Organizer
	Student Response System	X	Desktop Publishing		Image File(s)
	Web Design Software		Blog		Video
	Animation Software		Wiki		Electronic Game or Puzzle Maker
X	Email	X	Website		