



# GEORGIA

PEACH STATE PATHWAYS

Career, Technical, & Agricultural Education

## BUSINESS & COMPUTER SCIENCE

**PATHWAY:** Computing

**COURSE:** Beginning Programming

**UNIT:** 3.1 Algorithm Development and Programming Constructs



## INTRODUCTION

---

**Annotation:** Students are introduced to general programming constructs (statements, conditionals, variables, loops, etc.) and will be required to complete a program that will utilize each programming construct. During the unit, students will practice creating algorithms. Also, they will practice program design, including an introduction to CRC cards. Many teaching methods will be used throughout the unit, including: demonstration, lecture, independent work, and group work at teacher's discretion. Students will use the computer for productivity and possibly the Internet for research. This unit will take a few months to complete.

**Grade(s):**

|   |                  |
|---|------------------|
| X | 9 <sup>th</sup>  |
| X | 10 <sup>th</sup> |
| X | 11 <sup>th</sup> |
| X | 12 <sup>th</sup> |

**Time:** 12 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 pseudocode and/or flowcharts.
- b. Interpret algorithms expressed in pseudocode 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.

### **GPS Academic Standards:**

**SCSh3** Students will identify and investigate problems scientifically.

**MM3P1** Students will solve problems (using appropriate technology).

**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.

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

### **National Standards:**



# UNDERSTANDINGS & GOALS

---

## **Enduring Understandings:**

**Students should retain the understanding of each major programming construct. Additionally, students will develop algorithmic thinking to be used in the problem solving process. Finally, students will retain the ability to use CRC cards.**

## **Essential Questions:**

- What are conditional statements, iterative statements, and variables?
- How are programs documented and commented?
- What is an algorithm and what is it used for?
- What is a CRC card? How is a CRC card used?
- What is the relationship between classes?
- How can algorithms be modified to solve problems?

## **Knowledge from this Unit:**

- Students will demonstrate the ability to use conditional statements, iterative statements, and variables.
- Students will demonstrate the ability to properly document and comment programs.
- Students will use pseudocode and flowcharts to develop algorithms to solve simple problems.
- Students will demonstrate and interpret algorithms.
- Students will explain how CRC cards can be used to design classes.
- Students will identify and describe the relationship between classes.
- Students will analyze and modify algorithms to solve computational problems.

## **Skills from this Unit:**



# 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
- ☒ Objective assessment - multiple-choice, true- false, etc.
  - ☒ 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
- ☐ Subjective assessment/Informal observations
  - ☐ Essay tests
  - ☐ Observe students working with partners
  - ☐ 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:** Testing and Observation of Programming Techniques

**Assessment(s) Description/Directions:** Teachers should choose a method of formative evaluation to use during the unit. Both testing and student observations are methods that could possibly be used.

**Attachments for Assessment(s):**

**Web Resources:** <http://www.bluepelicanjava.com/>

**Web Resource Title:** Blue Pelican Java

**Web Resource Description:** A website with free resources can be used by programming teachers. Resources include: an e-textbook, assignments, projects, and tests that can be used to introduce students to Java programming.



# 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.

**BCS-BP-5** Students will design algorithms and programming solutions for a variety of computational problems.

**BCS-BP-6** Students will design solutions for simple programs using basic programming techniques and constructs.

**BCS-BP-7** Students will use and develop algorithms to solve simple problems.

**BCS-BP-8** Students will demonstrate knowledge of the relationships between classes.

### 2. Review Essential Questions.

- What are conditional statements, iterative statements, and variables?
- How are programs documented and commented?
- What is an algorithm and what is it used for?
- What is a CRC card? How is a CRC card used?
- What is the relationship between classes?
- How can algorithms be modified to solve problems?

### 3. Identify and review the unit vocabulary.

### 4. Assessment Activity.

(Based on a 50 minute period)

**Week 1:** Introduction to Algorithmic Thinking and Problem Solving

**Week 2:** Using Psuedocode and Flowcharts to express Algorithms

**Week 3:** Introduction to Programming

**Week 4:** Variables

**Weeks 5-6:** Conditional Statements

**Weeks 7-8:** Iteration

**Week 9:** Relationships between classes (using CRC cards)

**Week 10:** Putting it all together and developing testing plans

**Weeks 11-12:** Unit Performance Activity

**Attachments for Learning Experiences:** Please list.

## Notes & Reflections:



## CULMINATING PERFORMANCE TASK (Optional)

---

**Culminating Unit Performance Task Title:** BlackJack

**Culminating Unit Performance Task Description/Directions/Differentiated Instruction:** After covering the major themes of the unit, students will work individually to program a simple card game (BlackJack). Students should be required to complete all phases of program development, including: algorithm development, creation of CRC cards, implementation of necessary classes, and testing and debugging the program.

**Attachments for Culminating Performance Task:**

**Rubric for Performance Task:**

## **Java Project: Black Jack**

**Directions:** For your second project you will create a game. You will program Black Jack to be played when your program is executed. If you are unsure of how to play Black Jack below is a summary of what your program should be able to do.

The program should play the game of Black jack. At the beginning of each hand, the dealer (the program) and the player (the user) are each dealt two cards.

- If the player's cards total to 21, the hand is over and the player wins (unless the dealer also has 21, in which case the hand is a tie). The player may now draw additional cards in an attempt to get close to 21.
- If the player exceeds 21, the hand is over and the player loses. Otherwise, the dealer begins drawing cards until the player's total is beat or the dealer busts.
- If the dealer exceeds 21, the player wins.
- If the player's final count exceeds the dealer's final count, the player wins.
- If the dealer and the player have identical counts, the hand is a tie. Otherwise, the dealer wins.
- At the end of each hand, the program will display the total number of wins for both the dealer and the player. The player will then be asked whether or not to play another hand.
- Be sure to include one special element in the program (Ex: cheat code, betting, etc.)

You are required to produce the following for completion of the project:

1. Psuedocode or flowchart (develop an algorithm)
2. CRC Cards. (identify the classes and their relationships)
3. Functional Java program.
4. A testing plan.

Name: \_\_\_\_\_

| Task                                           | Possible Points | Points Earned |
|------------------------------------------------|-----------------|---------------|
| Algorithm developed using Psuedocode/Flowchart | 20              |               |
| CRC Cards                                      | 20              |               |
| Functional Programs meeting all specifications | 50              |               |
| A testing plan                                 | 10              |               |
| Total Points                                   | 100             |               |



# UNIT RESOURCES

---

## Web Resources:

## Attachment(s):

## Materials & Equipment:

Computer

Java Software Development Kit

Integrated Development Environment (Dr. Java, Blue J, etc.)

Microsoft PPT

Microsoft Word

Projector and screen

## What 21st Century Technology was used in this unit:

|                                     |                         |
|-------------------------------------|-------------------------|
| <input checked="" type="checkbox"/> | Slide Show Software     |
| <input type="checkbox"/>            | Interactive Whiteboard  |
| <input type="checkbox"/>            | Student Response System |
| <input type="checkbox"/>            | Web Design Software     |
| <input type="checkbox"/>            | Animation Software      |
| <input type="checkbox"/>            | Email                   |

|                                     |                    |
|-------------------------------------|--------------------|
| <input type="checkbox"/>            | Graphing Software  |
| <input checked="" type="checkbox"/> | Calculator         |
| <input checked="" type="checkbox"/> | Desktop Publishing |
| <input type="checkbox"/>            | Blog               |
| <input type="checkbox"/>            | Wiki               |
| <input checked="" type="checkbox"/> | Website            |

|                          |                                 |
|--------------------------|---------------------------------|
| <input type="checkbox"/> | Audio File(s)                   |
| <input type="checkbox"/> | Graphic Organizer               |
| <input type="checkbox"/> | Image File(s)                   |
| <input type="checkbox"/> | Video                           |
| <input type="checkbox"/> | Electronic Game or Puzzle Maker |