GEORGIA | PEACH STATE PATHWAYS

Career, Technical, & Agricultural Education

BUSINESS & COMPUTER SCIENCE

PATHWAY: Computing

COURSE: Intermediate Programming

UNIT: 2-Software Engineering, Programming, and Problem Solving

INTRODUCTION

Annotation:
Topics in this unit include software engineering, problems solving, user interface design, and programming. Methods include lecture, discussion, project based learning, demonstration, and independent practice. Not only will technology will be used to design user interfaces, but it will also be used to communicate with a client as a productivity tool in programming and documenting.

Grade(s):

☐ 9th
☐ 10th
☒ 11th
☐ 12th

Time: 16 weeks (based on a 50 minute class period)

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-IP-3 Students will demonstrate knowledge of key concepts in software engineering.

a. Define software engineering.

b. Compare and contrast software engineering and programming.

c. List the phases in the software life cycle.

d. Perform software requirements analysis.

e. Discuss Extreme Programming and pair programming.

f. Discuss societal and ethical issues in software engineering.

BCS-IP-5 The student will collaboratively develop solutions for specific problems.

a. Collaboratively determine a course of action for problem resolution.

b. Design algorithms for problem resolution.

c. Break a task into subtasks required for problem resolution.

d. Select appropriate tools and technology resources to accomplish a variety of tasks.

e. Collaboratively design, combine, test, analyze, and adjust coding solutions based on problem-solving algorithms.

f. Review and discuss coding solutions for elements of thoroughness and correctness.

BCS-IP-6 Students will design solutions for complex programs using advanced programming techniques and constructs.

a. Implement techniques such as conditional statements, iterative statements, and variables to solve complex problems.

b. Utilize basic and advanced mathematical expressions to solve complex problems.

c. Create appropriate arrays and lists.

d. Utilize various testing and debugging techniques to test classes.

e. Design classes that can be used in other programs.

f. Analyze and explain simple programs involving advanced programming constructs.

BCS-IP-7 Students will use and develop algorithms to solve complex problems.

a. Develop algorithms to solve complex problems using pseudocode.

b. Interpret algorithms written in pseudocode to code solutions to complex problems.

c. Identify properties of well-written algorithms in solving complex problems.

d. Use an Action, Components, and Events (ACE) chart to design your GUI components.

BCS-IP-8 Students will demonstrate knowledge of advanced object-oriented concepts.

a. Define polymorphism, interface, inheritance, encapsulation, and abstract class.

b. Develop programs that use inheritance and interfaces in the development of GUI applications.

GPS Academic Standards:

ELA12W3 The student uses research and technology to support writing.

ELA12LSV1 The student participates in student-to-teacher, student-to-student, and group verbal interactions.

ELA12W1 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.

ELA12W2 The student demonstrates competence in a variety of genres.

SCSh3 Students will identify and investigate problems scientifically.

National Standards:

UNDERSTANDINGS & GOALS

Enduring Understandings:

- At the conclusion of this unit, students will understand the process of software engineering and the relationship between software engineering and programming. Additionally, students will be able to design programs to accomplish complex tasks. Students will understand how to program using advanced programming constructs. Finally, students will be able to communicate properly with a client.

Essential Questions:

- What is software engineering?
- What is the difference between software engineering and programming?
- What is a software development life cycle?
- What are alternatives to the software development life cycle?
- How are existing programs analyzed and modified?
- How are algorithms developed to solve problems?
- What are advanced object oriented concepts?

Knowledge from this Unit:

- Students will be able to define software engineering.
- Students will be able to compare and contrast software engineering and programming.

- Students will be able to describe and apply the software development life cycle (including the different versions).
- Students will be able to identify alternatives to the software development life cycle.
- Students will be able to implement advanced programming techniques.
- Students will analyze and modify existing programs.
- Students will develop original algorithms to solve simple problems.
- Students will be able to explain and apply advanced object oriented concepts.

Skills from this Unit:

- Students will display appropriate interpersonal communication skills.

  Students will write using technology.

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
__X__ 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
_x_ 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
__X__ Constructed Responses
__ Chart good reading/writing/listening/speaking habits
__ Application of skills to real-life situations/scenarios
_____ Post-test

Assessment(s) Title: **Constructed Response, Self Evaluation and Informal Checks**

Assessment(s) Description/Directions: A variety of assessments should be used throughout the unit. Students should be required to complete programs that build programming skills. Teacher should monitor and facilitate students through the duration of the project. Students will also be required to perform a self assessment.

Attachments for Assessment(s):

Self Evaluation:

Self-Evaluation for Working with a Client Project

Name:_______________________

Project Name:_____________________

Client Name:_______________________

1. What was the most challenging aspect of this project for you?

2. What do you feel that you did best during the project?

3. List three things you learned from this project.

4. In terms of communicating with the client, what was the most difficult aspect?

5. What advice would you provide to students completing this project in the future?

Please list any other comments below:

Web Resource: http://www.bluepelicanjava.com/

Web Resource Title: Blue Pelican Java

Web Resource Description: A resource (free e-book) for teaching advanced programming techniques.

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 software engineering?
    - What is the difference between software engineering and programming?
    - What is a software development life cycle?
    - What are alternatives to the software development life cycle?
    - How are existing programs analyzed and modified?
    - How are algorithms developed to solve problems?
    - What are advanced object oriented concepts?
3. **Identify and review the unit vocabulary.**
4. **Assessment Activity.**

Sequence of Instruction and Learning:
(Based on a 50 minute class period)
Weeks 1 - 2: Introduction to Software Engineering, Software Development Life Cycle
Weeks 3 - 6: Advanced Programming Techniques
Weeks 7 - 8: Developing Algorithms for Complex Problems
Weeks 9 - 12: Advanced Object Oriented Programming Concepts
Weeks 13 - 16: Unit Performance Task

Technology Connection/Integration
Technology is used by students to create a program and properly document the process the student went through. Additionally, students will use technology for desktop publishing (interface development). Students will also use technology to communicate with group members and the client. Finally, students will use technology as a productivity and problem solving tool when developing programs and algorithms. Teachers will use technology to present information and monitor student progress.

Attachments for Learning Experiences: Please list.

Notes & Reflections:

- This should be the final unit of the course and used as a culminating project for the course and Computing Pathway. For the purposes of this project, the teacher should find clients (teachers, clubs, advisors, coaches, etc.) throughout the school. Students will then choose the project and client they would like to work with on the project.

CULMINATING PERFORMANCE TASK (Optional)

Culminating Unit Performance Task Title: Working with a Client Project

Culminating Unit Performance Task Description/Directions/Differentiated Instruction:

Prior to implementing the unit performance task, three things should be completed. First, students should be introduced to the advanced programming techniques, object-oriented programming, and constructs. Sufficient practice time should be given to students. Second, students should have a sound understanding of software engineering and the software development life cycle. Third, and perhaps the most important, the teacher should contact other teachers and find possible projects and clients to work with for the project.

Students may work in pairs or groups of three. Each student should have a role and title in the project. Release time must be given for students to meet with their clients. This project could take as few as two weeks and as many as four weeks.

Specifications and rubric are attached. An evaluation form for the client and a form for the student are attached.

Attachments for Culminating Performance Task:

Rubric for Performance Task:

Working with a Client Project

Directions: For this project you will have the opportunity to build a program for a client within the school. I have contacted teachers that need a program built for use in their classrooms. Choose one program from the provided list. Over the next several weeks you will meet with the client at least twice a week and go through all of the steps necessary to create a useable, deliverable and functional program for that teacher. Basically, you will choose one model of the software development life cycle and follow it throughout the entire process.

Along the way you will need to document the following. At the conclusion of this project you will need to turn in a folder/three ring binder with the following options:

1. **Project description:** a general description of the project (50 words or less). In this description please identify which Software Development Model you will follow.
2. **Prototypes of User Interface Design (screenshots):** simple printouts of the screen shots. Client must sign off on these.
3. **Actual Program Code (printed):** the code of the actual program. The code should be printed off and submitted with your folder/three ring binder.
4. **Program code (files):** the actual files that make up your program.
5. **Bi-weekly updates:** every Tuesday and Thursday you will update your progress by writing a 200 word journal entry.
6. **Client Evaluation (attached):** your client will evaluate your work and the communication skills you used throughout the semester.
7. **Reflection:** a self-reflection on your performance and what you learned by completing the project.

Additional Document:

Client Evaluation for Working with a Client Project

Student Name:________________________

Project Name:________________________

Client Name:_________________________

Directions: Please evaluate the student on the following criteria.

1. Student communicated clearly throughout the semester.

Needs Improvement   Acceptable   Excellent
1   2   3   4   5

2. Student provided a prototype for your approval.

Needs Improvement   Acceptable   Excellent
1   2   3   4   5

3. Student produced a usable and functional program.

Needs Improvement		Acceptable		Excellent
1	2	3	4	5

4. Student provided training on how to use the program.

Needs Improvement		Acceptable		Excellent
1	2	3	4	5

5. Student made changes to program based on your feedback.

Needs Improvement		Acceptable		Excellent
1	2	3	4	5

Other comments:

Client Signature: ________________________

Date: ________

UNIT RESOURCES

Web Resources:

Attachment(s):

Materials & Equipment:

Computer

Internet connection

Network Storage Space

Three ring folder/binder

Desktop Publishing Software

Microsoft PowerPoint and Word

Java Software Development Kit (5.0)-suggested

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

Projector and screen

Interactive Whiteboard

What 21st Century Technology was used in this unit:

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