AP Java

Revised UBD Curriculum

Egg Harbor Township High School

Business and Computer Science Department

CTE

Career and Technical Education

Created By: Adam Swift

Coordinated By: Dr. Carmelita Graham

June 2016

DISTRICT MISSION STATEMENT

Our mission in the Egg Harbor Township School District is to partner with the student, family, school, and community to provide a safe learning environment that addresses rigorous and relevant 21st Century standards and best practices which will develop academic scholarship, integrity, leadership, citizenship, and the unique learning style of students, while encouraging them to develop a strong work ethic and to act responsibly in their school community and everyday society.

CAREER AND TECHNICAL EDUCATION

Mission:

New Jersey's Office of Career and Technical Education seeks to prepare students for career opportunities of the 21st century, succeed as global citizens and support healthy economic growth for New Jersey. Career and Technical Education prepares students to succeed as global citizens for career opportunities for the 21st Century and to support healthy economic growth within the state.

INTRODUCTION

The most precious resource teachers have is time. Regardless of how much time a course is scheduled for, it is never enough to accomplish all that one would like. Therefore, it is imperative that teachers utilize the time they have wisely in order to maximize the potential for all students to achieve the desired learning.

High quality educational programs are characterized by clearly stated goals for student learning, teachers who are well-informed and skilled in enabling students to reach those goals, program designs that allow for continuous growth over the span of years of instruction, and ways of measuring whether students are achieving program goals.

EGG HARBOR TOWNSHIP SCHOOL DISTRICT CURRICULUM TEMPLATE

The Egg Harbor Township School District has embraced the backward-design model as the foundation for all curriculum development for the educational program. When reviewing curriculum documents and the Egg Harbor Township curriculum template, aspects of the backward-design model will be found in the stated enduring *understandings/essential questions, unit assessments,* and *instructional activities*. Familiarization with backward-design is critical to working effectively with Egg Harbor Township's curriculum guides.

GUIDING PRINCIPLES: WHAT IS BACKWARD DESIGN?

WHAT IS UNDERSTANDING BY DESIGN?

“Backward design” is an increasingly common approach to planning curriculum and instruction. As its name implies, “backward design” is based on defining clear goals, providing acceptable evidence of having achieved those goals, and then working ‘backward’ to identify what actions need to be taken that will ensure that the gap between the current status and the desired status is closed.

Building on the concept of backward design, Grant Wiggins and Jay McTighe (2005) have developed a structured approach to planning programs, curriculum, and instructional units. Their model asks educators to state goals; identify deep understandings, pose essential questions, and specify clear evidence that goals, understandings, and core learning have been achieved.

Program based on backward design use desired results to drive decisions. With this design, there are questions to consider, such as: What should students understand, know, and be able to do? What does it look like to meet those goals? What kind of program will result in the outcomes stated? How will we know students have achieved that result? What other kinds of evidence will tell us that we have a quality program? These questions apply regardless of whether they are goals in program planning or classroom instruction.

The backward design process involves three interrelated stages for developing an entire curriculum or a single unit of instruction. The relationship from planning to curriculum design, development, and implementation hinges upon the integration of the following three stages.

Stage I: Identifying Desired Results: **Enduring understandings, essential questions, knowledge and skills need to be woven into curriculum publications, documents, standards, and scope and sequence materials. Enduring understandings identify the ‘ ‘big ideas’ ’ that students will grapple with during the course of the unit. Essential questions provide a unifying focus for the unit and students should be able to answer more deeply and fully these questions as they proceed through the unit. Knowledge and skills are the** ***‘ ‘stuff’ ’*** upon which the understandings are built.

Stage II: Determining Acceptable Evidence: Varied types of evidence are specified to ensure that students demonstrate attainment of desired results. While discrete knowledge assessments (e.g.: multiple choice, fill-in-the-blank, short answer, etc…) will be utilized during an instructional unit, the overall unit assessment is performance-based and asks students to demonstrate that they have mastered the desired understandings. These culminating (summative) assessments are authentic tasks that students would likely encounter in the real-world after they leave school. They allow students to demonstrate all that they have learned and can do. To demonstrate their understandings students can explain, interpret, apply, provide critical and insightful points of view, show empathy and/or evidence self-knowledge. Models of student performance and clearly defined criteria (i.e.: rubrics) are provided to all students in advance of starting work on the unit task.

Stage III: Designing Learning Activities: Instructional tasks, activities, and experiences are aligned with stages one and two so that the desired results are obtained based on the identified evidence or assessment tasks. Instructional activities and strategies are considered only once stages one and two have been clearly explicated. Therefore, congruence among all three stages can be ensured and teachers can make wise instructional choices.

At the curricular level, these three stages are best realized as a fusion of research, best practices, shared and sustained inquiry, consensus building, and initiative that involves all stakeholders. In this design, administrators are instructional leaders who enable the alignment between the curriculum and other key initiatives in their district or schools. These leaders demonstrate a clear purpose and direction for the curriculum within their school or district by providing support for implementation, opportunities for revision through sustained and consistent professional development, initiating action research activities, and collecting and evaluating materials to ensure alignment with the desired results. Intrinsic to the success of curriculum is to show how it aligns with the overarching goals of the district, how the document relates to district, state, or national standards, what a high quality educational program looks like, and what excellent teaching and learning looks like. Within education, success of the educational program is realized through this blend of commitment and organizational direction.

INTENT OF THE GUIDE

This guide is intended to provide teachers with course objective and possible activities, as well as assist the teacher in planning and delivering instruction in accordance with the New Jersey Core Curriculum Content Standards. The guide is not intended to restrict or limit the teacher's resources or individual instruction techniques. It is expected that the teacher will reflectively adjust and modify instruction and units during the course of normal lessons depending on the varying needs of the class, provided such modified instruction attends to the objectives and essential questions outlined below.

Unit Name: Programming Basics Review
Time Frame: Approximately 4 weeks
Author: Adam Swift

UNIT

Subject: Computer Science Country: USA

Course/Grade: AP Computer Science (10-12) State/Group: NJ

School: Egg Harbor Twp High School

UNIT SUMMARY

Students will reinforce their understanding of all material from the Intro course including but not limited to: basic syntax/semantics, variables, selection statements, loops, conditions, Boolean Algebra, and developing classes. The focus of this unit will be on being able to apply knowledge from the introductory course in a manner that is required on the AP Computer Science Exam. Students will demonstrate this skill by completing a variety of review questions/packets/programs from different AP Test Prep Materials.

UNIT RESOURCES

Textbook – Fundamentals of Java Chapter 1-4
Textbook – Java Software Solutions AP Computer Science Chapter 1-3
Website – PreLab Projects, Post Chapter Questions and Projects
Planning Posterboard (mapping programs)
AP Computer Science Barron's Test Review Book
College Board AP Computer Science Webpage
Mini-Whiteboards: Tracing Programs
Software/Compiler – JGrasp & BlueJ
Computers
Pens, Pencils, Markers, & Expo Markers
Internet Access

Internet Resource Links:

http://homes.cs.washington.edu/~reges/teals/
http://homes.cs.washington.edu/~reges/teals/bjp.shtml
http://www.skylit.com/beprepared/fr.html
http://apcentral.collegeboard.com/apc/public/exam/exam_information/2000.html
https://codehs.com/
http://chortle.ccsu.edu/CS151/cs151java.html
http://csunplugged.org/activities/
https://users-mooc.amplify.com/apcs
www.phschool.com

STAGE ONE

GOALS AND STANDARDS

Students will be able to answer questions and program at a level consistent with AP Exam expectations. They will be able to take their understanding of introductory material and apply it to answer AP Style Multiple Choice and Open Ended Questions. By the end of the unit students should be able to trace and *explain other's code as well as create their own programs using variables, selection statements, loops, conditions, Boolean Algebra, and classes.*

21st Century.9.CPR2 Apply appropriate academic and technical skills
21st Century.9.CPR6 Demonstrate creativity and innovation.
21st Century.9.CPR11 use technology to enhance productivity
8.2.12.D.3 Determine and use the appropriate resources (e.g., CNC (Computer Numerical Control) equipment, 3D printers, CAD software) in the design, development and creation of a technological product or system.
8.2.12.E.1 Demonstrate an understanding of the problem-solving capacity of computers in our world.
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).
TEC.9-12.9.4.12.K.(4).5 - [Cumulative Progress Indicator] - Use the software development process to design a software application and deliver it to the customer.
ITEC.9-12.9.4.12.K.(4).7 - [Cumulative Progress Indicator] - Implement software testing procedures to ensure quality products.
STEM.9-12.9.4.12.O.(2).6 - [Cumulative Progress Indicator] - Demonstrate the knowledge and technical skills needed to obtain and succeed in a chosen scientific and mathematical field.
TEC.9-12. - [Content Statement] - The design process is a systematic approach to solving problems
TEC.9-12.8.2.12.F.1 - [Cumulative Progress Indicator] - Determine and use the appropriate application of resources in the design, development, and creation of a technological product or system.
12.9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications.
12.9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.
12.9.3.IT-PRG.6 Program a computer application using the appropriate programming language.

ENDURING UNDERSTANDING

Students will be able to follow Boolean logic in programs understanding that order of execution is not always top to bottom. Students will understand that they can gain control of their programs, improving efficiency, by including these concepts in their own programs. Students will also be able to use their programs in slightly different ways due to implementation of conditions allowing them to skip certain code when necessary. Once again they will value the ability to type multi-purpose programs rather than several mini stand alone programs that do the same thing with just one minor difference. Efficiency is essential in technology when we consider the cost of storage capacity and processing power. It is better to create efficient code than redundant code and excessive mini-programs.

ESSENTIAL QUESTIONS

Why is efficiency important when using technology and programming?
Why is succinct and dynamic coding important?
Why and under what scenarios would programmers want to maintain control of their programs even after code development?
How can we create effective code that works under several scenarios and processes multiple times without being repetitive?
How does an understanding of object oriented programming, regardless of language, prepare you for success in Computer Science?

How is computer science used in today's world and are you capable of correcting, modifying, and creating mini programs?

KNOWLEDGE AND SKILLS

Students will be able to identify, understand, and apply concepts associated with variables, selection statements, and control statements

- Understand the creation and use of variables
- Understand conditionals and how they execute based on Boolean logic
- Understand how to create complex, multi-faceted conditions
- Understand/Follow Decision Trees
- Understand the difference between selection statements and repetition statements
- Understand how to follow the execution order of control statements in other programmers' code
- Trace code using conditionals and loops
- Make decisions in their programs based on user input
- Repeat lines of code efficiently by using loops
- Understand the importance of control statements in improving the dynamic nature and efficiency of their code
- Create programs that work under various scenarios
- Be able to identify what control statements are appropriate at given times
    - Decision making statements
    - Nested decision
    - Loops

STAGE TWO

PERFORMANCE TASKS

After review of PPT's and classroom dicussions students will answer review questions pertaining to basic syntax/semantics, variables, selection statements, loops, conditions, Boolean Algebra, and classes. Questions will test the students understanding of material and require them to apply concepts in not only creating their own programs, but, analyzing the programs of others. Students will use their analytical skills in answering AP Style Multiple Choice Questions from a variety of AP Exam Prep books.

Students will trace completed programs from the text. In order to demonstrate their mastery of these introductory concepts, students will trace the execution of other's code by typing completed programs and adding comments explaining the code's execution. This will reinforce proper syntax while also forcing them to explain how these concepts work.

Students will also be required to create their own programs using these concepts. They will have to demonstrate they know the proper structure of a Java program. Students will identify program requirements and create efficient programs to meet these demands. Additionally, students will attempt to answer AP Style Open Ended Questions (from previous years' AP Exams) requiring them to understand current code and modify it accordingly. This will allow them to become familiar with the format and structure of Open Ended questions on the AP Exam. It will also demonstrate the expectations for passing the AP Exam.

OTHER EVIDENCE

Quizzes/Test

Completion of assigned worksheets/questions
Classroom discussion and interaction during activities
Completion of programming projects using Java Language
Completion of Chapters in AP Exam Review Books (Barron's)
Question of the Day Packet (AP Style Multiple Choice Questions)
Completion of AP Style Open Ended Questions (previous AP Exam test questions)

STAGE THREE

LEARNING PLAN

Java Syntax and Semantics – How to create simple Java Programs and follow the execution of other's code

- Review of AP Computer Science Summer Work
  - Demo of Programming Projects
  - Discussion/Correction of Review Handouts

- Completion of Opening of Year Programming Packet
  - Slot Machine
  - CD Investment
  - Dice Game

- Worksheets form University of Washington Computer Science & Engineering: TEALS Workshop
  - http://homes.cs.washington.edu/~reges/teals/

- Videos that should be viewed in advance of classroom lessons (certain assignments – flip classroom)
  - http://homes.cs.washington.edu/~reges/teals/bjp.shtml

- Online Quizzes and Flash Cards to Monitor Student progress throughout unit
  - http://chortle.ccsu.edu/CS151/cs151java.html

- Online practice questions – codehs.org

- AP Computer Science Exam Review Books
  - Chapters: variables, Boolean Algebra, selection statements, repetition statements, and conditionals

- Additional Program Possibilities:
  http://homes.cs.washington.edu/~reges/teals/bjp.shtml
  www.phschool.com
  Instructor Resources CD – Fundamental of Java AP Computer Science Essentials (textbook)

Unit Name: Enhancing Classes
Time Frame: Approximately 8 weeks
Author: Adam Swift

UNIT

Subject: Computer Science Country: USA

Course/Grade: AP Computer Science (10-12) State/Group: NJ

School: Egg Harbor Twp High School

UNIT SUMMARY

With an understanding of objects and object oriented programing, students will start to better organize their code by creating classes to develop objects that perform services they define. This unit will further delve into the details of class definitions including the structure and semantics of methods as well as the scope and encapsulation of data. Students will be introduced to Null References, This statements, Aliases, Parameter Passing Issues, and interfaces. They will further investigate how driver programs interact with classes and the limitations/impact of this interaction.

UNIT RESOURCES

Textbook – Fundamentals of Java Chapter 5-6
Exam View Assessment Suite Chapter 5-6
Instructor Resources CD Chapter 5-6 (PPT, Handouts, Solutions)
Textbook – Java Software Solutions AP Computer Science Chapter 4 & 5
Website – PreLab Projects, Post Chapter Questions and Projects
AP Computer Science Barron's Test Review Book
College Board AP Computer Science Webpage
Planning Posterboard (mapping programs)
Mini-Whiteboards: Tracing Programs
Software/Compiler – JGrasp & BlueJ
Computers
Pens, Pencils, Markers, & Expo Markers
Internet Access

Internet Resource Links:

http://homes.cs.washington.edu/~reges/teals/
http://homes.cs.washington.edu/~reges/teals/bjp.shtml
http://www.skylit.com/beprepared/fr.html
http://apcentral.collegeboard.com/apc/public/exam/exam_information/2000.html
https://codehs.com/
http://chortle.ccsu.edu/CS151/cs151java.html
http://csunplugged.org/activities/
https://users-mooc.amplify.com/apcs
www.phschool.com

STAGE ONE

GOALS AND STANDARDS

Students will be able to develop classes from which they can create objects. They will learn about the different components of classes: data declarations and method declarations. They will also understand that once they have created the class (the blueprint), an outside program can create objects from the class and interact with the methods/components of the class through this object. Classes will be culmulative including all the previous concepts taught throughout the previous year. Students will be introduced to more complex issues with classes including Null References, This statements, Aliases, Parameter Passing Issues, and Interfaces. Students will understand the value of classes in further improving efficiency and code organization. Students will see how one class can be utilized in a variety of different programs with each using objects from the class differently. This concept of writing classes and implementing it later is the foundation of Object Oriented Programming. Taking their previous understanding of creating classes and adding the new enhancements will allow students to create highly functional and dynamic classes. Students will also be able to analyze and interact with classes created by other programmers.

21st Century.9.CPR2 Apply appropriate academic and technical skills
21st Century.9.CPR6 Demonstrate creativity and innovation.
21st Century.9.CPR11 use technology to enhance productivity
8.2.12.D.3 Determine and use the appropriate resources (e.g., CNC (Computer Numerical Control) equipment, 3D printers, CAD software) in the design, development and creation of a technological product or system.
8.2.12.E.1 Demonstrate an understanding of the problem-solving capacity of computers in our world.
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).
TEC.9-12.9.4.12.K.(4).5 - [Cumulative Progress Indicator] - Use the software development process to design a software application and deliver it to the customer.
ITEC.9-12.9.4.12.K.(4).7 - [Cumulative Progress Indicator] - Implement software testing procedures to ensure quality products.
STEM.9-12.9.4.12.O.(2).6 - [Cumulative Progress Indicator] - Demonstrate the knowledge and technical skills needed to obtain and succeed in a chosen scientific and mathematical field.
TEC.9-12. - [Content Statement] - The design process is a systematic approach to solving problems
TEC.9-12.8.2.12.F.1 - [Cumulative Progress Indicator] - Determine and use the appropriate application of resources in the design, development, and creation of a technological product or system.
12.9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications
12.9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.
12.9.3.IT-PRG.6 Program a computer application using the appropriate programming language

ENDURING UNDERSTANDING

Writing classes is at the foundation of Object Oriented Programming. Up to this point we have used other programmers' classes for the purpose for which they created them. Now we will be able to create our own classes from which we and others can develop objects and interact with our code. This improves our efficiency by limiting redundant code. It is also what object oriented programming is all about. This *is the beginning of illustrating how open source programming works – allowing outsiders to use your code, but, controlling the scenrios in which they can.*

ESSENTIAL QUESTIONS

What is object oriented programming and how does it allow for better organized and understandable code?
Understanding the role of programming/technology in today's world, why is organization and standrard structure necessary when programming?
What is a class in Java and, although it does not run, why do we write them?
How can we create effective code that works under several scenarios without being repetitive?
How does an understanding of object oriented programming, regardless of language, prepare you for success in Computer Science?
How is computer science used in today's world and are you capable of correcting, modifying, and creating mini programs?

KNOWLEDGE AND SKILLS

Students will be able to Understand and Create Classes

- Understand the value of creating classes even though they do not run
- Understand Classes as blueprints that other programs will interact with
- Understand the components of Classes
- Understand the structure of a Class
- Be able to trace the execution of driver prorams using different classes, including being able to follow the jumping back and forth between classes
- Be able to create their own classes with the necessary encapsulated data and methods accessible to outsiders
- Understand scope in terms of what can and can't be accessed and used in their classes
- Understand how to call constructors to create objects from classes giving them access to other information within that class
- Understand and implement overloaded and overridden methods by understanding method signatures
- Create methods with correct method headers (return types and parameters) as well as method bodies that act almost like mini programs
- Understand how to enhance classes through an understanding of Null References, This Statements, Parameter Passing, and Interfaces
- Create driver programs that illustrate how classes work

STAGE TWO

PERFORMANCE TASKS

After review of PPT's and classroom dicussions students will answer textbook review questions pertaining to Enhancing Classes: Null References, This statements, Aliases, Parameter Passing Issues, and Interfaces. This information will build on their previous understanding of data declarations & method declarations, encapsulation, classes as blueprints, constructors, using methods with parameters, format of methods in classes (proper header and method body), and driver programs. Questions will test the students understanding of classes whether it is developing them, tracing execution, or interacting correctly with them.

Students will trace completed programs from the text. In order to become familiar with the syntax of Classes in Java and practice mapping out the execution, students will type completed programs (both driver program and class itself), adding comments explaining the code's execution. Students will also be be able to answer a variety of AP style Multiple Choice Questions pertaining to classes. These questions will involve interacting with/analyzing commonly used classes (String Class, Math Class, etc) as well as newly defined classes. In answering these questions students will be required to apply their understanding of Classes in an analytical manner.

Students will be required to create classes and driver programs that use them. They will have to demonstrate they know the proper structure of classes as well as which things should be accessible and which should not be. They will also show the ability to interact with classes that they have created as well as classes others have created. Students will use material covered throughout preceding chapters in formulating their classes. This process will be enhanced through the completion of previous year AP Open Ended Questions from the AP Computer Science Exam. These questions will require the ability to analyze previously developed code and knowledge of how to work with this code and add components to it.

OTHER EVIDENCE

Quizzes/Test
Completion of assigned worksheets/questions
Classroom discussion and interaction during activities
Completion of programming projects using Java Language
Completion of Chapters in AP Review Books (Barron's)
Question of the Day Packet (AP Style Multiple Choice Questions)
Completion of AP Style Open Ended Questions (previous AP Exam test questions)

STAGE THREE

LEARNING PLAN

Java Syntax and Semantics – rules and structure for Classes (creating and enhancing)

- *Fundamentals of Java:* Chapter 5-6 Powerpoint
  - Pre-Lab Activies from phschool.com website (pre-test: what do we know?)
  - Tracing/Commenting out of sample programs throughout chapter to explain program's execution
- *Fundamentals of Java:* End of Section and Chapter review questions throughout text in Chapters 45-6
- *Java Software Solutions for AP Computer Science A:* end of chapter review questions (Chapter 4 & 5)
  - phschool.com website: Tracing References and Parameter Passing Activities
- Worksheets form University of Washington Computer Science & Engineering: TEALS Workshop
  - http://homes.cs.washington.edu/~reges/teals/
- Videos that should be viewed in advance of classroom lessons (certain assignments – flip classroom)
  - http://homes.cs.washington.edu/~reges/teals/bjp.shtml
- Online Quizzes and Flash Cards to Monitor Student progress throughout unit
  - http://chortle.ccsu.edu/CS151/cs151java.html
- Online practice questions – codehs.org

Modifying and Creating Interactive Programs with Variables and User Input

- Students will create classes using JGrasp & BlueJ

- Sample Programming Project:
    - Create a class that represents an employee. Every employee should have a name, job title, and hourly rate all of which will be defined in the constructor. Your class should include methods that will get access to each of the instance variables. You should also have a method that accepts one parameter that represents the hours worked in a given week and calculates the pay for that person based off of that amount of hours (make sure to consider overtime, any hours over 40 1.5*pay). You also should include a method that accepts one parameter to represent an employee rating (1 = poor, 2 = good, 3 = outstanding) and then determines the raise for that employee creating a new hourly rate. This method should not return anything. Additionally, your class should have a method that prints the employee's name, job title, and hourly rate. Once your class is finished, you should create a tester program that checks to see that your methods work.

- AP Computer Science Open Ended Questions – all previous test questions can be found online
  http://www.skylit.com/beprepared/fr.html
  http://apcentral.collegeboard.com/apc/public/exam/exam_information/2000.html

- Additional Program Possibilities:
  http://homes.cs.washington.edu/~reges/teals/bjp.shtml
  www.phschool.com
  Instructor Resources CD – Fundamental of Java AP Computer Science Essentials (textbook)

Unit Name: Arrays & ArrayLists
Time Frame: Approximately 8 weeks
Author: Adam Swift

UNIT

Subject: Computer Science Country: USA

Course/Grade: AP Computer Science (10-12) State/Group: NJ

School: Egg Harbor Twp High School

UNIT SUMMARY

Students will improve the efficiency of their programs in regards to data storage and processing powers. They will be introduced to Arrays and ArrayLists as a way of grouping data into lists that can be manipulated and organized accordingly. Students will understand Arrays and ArrayLists including, but not limited to, fundamental differences between the two and how to select the appropriate one, how to create and fill each, what type of "things" each can contain as well as how the type of data in the Array/ArrayLists effects behaviors, and how to manipulate included data. Students will also work with 2D Arrays understanding how to create and manipulate these. In working with Arrays (one and two-dimensional) and ArrayLists students will revisit common programming concepts such as variables, loops, selection statements, and parameter passing.

UNIT RESOURCES

Textbook – Fundamentals of Java Chapter 10 & 12
Exam View Assessment Suite Chapter 10 & 12
Instructor Resources CD Chapter 10 & 12 (PPT, Handouts, Solutions)
Textbook – Java Software Solutions AP Computer Science Chapter 6
Website – PreLab Projects, Post Chapter Questions and Projects
AP Computer Science Barron's Test Review Book
College Board AP Computer Science Webpage
Planning Posterboard (mapping programs)
Mini-Whiteboards: Tracing Programs
Software/Compiler – JGrasp & BlueJ
Computers
Pens, Pencils, Markers, & Expo Markers
Internet Access

Internet Resource Links:

http://homes.cs.washington.edu/~reges/teals/
http://homes.cs.washington.edu/~reges/teals/bjp.shtml
http://www.skylit.com/beprepared/fr.html
http://apcentral.collegeboard.com/apc/public/exam/exam_information/2000.html
https://codehs.com/
http://chortle.ccsu.edu/CS151/cs151java.html
http://csunplugged.org/activities/
https://users-mooc.amplify.com/apcs
www.phschool.com

STAGE ONE

GOALS AND STANDARDS

Students will understand how to create, fill, and interact with Arrays. They will understand Arrays as objects, but yet, be able to deal with individual elements of an array accordingly based on their types. Students will also be able to create arrays filled with objects from classes they developed, building on their understanding of classes. Students will work with ArrayLists, a class already defined in the Java Library. They will understand that when working with ArrayLists they must use the methods of the predefined class. Additionally, students will be able to identify when the use of a two-dimensional array is appropriate and be able to develop them. Students will be able to use Arrays and ArrayLists in their programs and answer multiple choice questions requiring analyzing code using these concepts.

21st Century.9.CPR2 Apply appropriate academic and technical skills
21st Century.9.CPR6 Demonstrate creativity and innovation.
21st Century.9.CPR11 use technology to enhance productivity
8.2.12.D.3 Determine and use the appropriate resources (e.g., CNC (Computer Numerical Control) equipment, 3D printers, CAD software) in the design, development and creation of a technological product or system.
8.2.12.E.1 Demonstrate an understanding of the problem-solving capacity of computers in our world.
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).
TEC.9-12.9.4.12.K.(4).5 - [Cumulative Progress Indicator] - Use the software development process to design a software application and deliver it to the customer.
ITEC.9-12.9.4.12.K.(4).7 - [Cumulative Progress Indicator] - Implement software testing procedures to ensure quality products.
STEM.9-12.9.4.12.O.(2).6 - [Cumulative Progress Indicator] - Demonstrate the knowledge and technical skills needed to obtain and succeed in a chosen scientific and mathematical field.
TEC.9-12. - [Content Statement] - The design process is a systematic approach to solving problems
TEC.9-12.8.2.12.F.1 - [Cumulative Progress Indicator] - Determine and use the appropriate application of resources in the design, development, and creation of a technological product or system.
12.9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications
12.9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.
12.9.3.IT-PRG.6 Program a computer application using the appropriate programming language

ENDURING UNDERSTANDING

Arrays and ArrayLists are essential for efficient programming, however, they do not come without challenges. Although entire Arrays/ArrayLists are objects, one must take note of the type of data contained within them and interact with the elements accordingly. A strong programmer will be able to identify the practical uses of Arrays/ArrayLists. They also will be able to avoid the pitfalls associated with confusing the concept of Arrays are objects and should be treated like so, with elements of arrays being treated appropriate per their type. Students will come to appreciate the effectiveness of arrays while understanding the detail in which they must be handled.

ESSENTIAL QUESTIONS

Why is efficient programming with limited variables important in the real world?
What is an Array/ArrayLists and why do we use them in our programs?
How do you create an Array in your program and interact with that Array?
How can we identify when to use an Array vs an ArrayLists?
Why is it important to be able to use concepts in our programs as well as analyze/interpret the code of other programmers?
How does an understanding of object oriented programming, regardless of language, prepare you for success in Computer Science?
How is computer science used in today's world and are you capable of correcting, modifying, and creating mini programs?

KNOWLEDGE AND SKILLS

Students will be able to understand, create, and manipulate Arrays (one and two-dimensional) & ArrayLists

- Understand the value of creating Arrays and ArrayLists
- Understand arrays as objects and elements as individual components that should be treated accordingly
- Understand the role of loops in creating and processing through Arrays/ArrayLists
- Understand the difference between indexes and elements
- Identify when to use either an Array or ArrayList
- Be able to trace the use of Arrays/ArrayLists by other programmers
- Be able to create Arrays filled with objects and then interact with individual objects from that array identifying the methods available to them
- Create two dimensional arrays using nested for loops and access individual elements from those arrays in row major order
- Understand how to interact with the ArrayLists Class including improving our storage efficiency through the use of the Add and Remove methods of the class
- Identify the diference between passing an entire array as a parameter and passing an individual element of the array (consider whether an alias is being created or not)
- Understand and be able to use Enhanced For Loops – understand that they can only be used to process through an already filled array not to fill one due to the fact that you are not referring to individual indexes. You are extracting the actual object in a specified index starting at 0 and ending at length-1
- Understand how to use sorting and searching algorithms when working with Arrays

STAGE TWO

PERFORMANCE TASKS

After review of PPT's and classroom dicussions students will answer textbook review questions pertaining to Arrays (one and two-dimensional) and ArrayLists. Students will create graphic orgainzers to identify the fundamental differences between Arrays and ArrayLists including when and how to use each. In working with arrays students will reinforce concepts of variables, loops, and object oriented programming through the inclusion of objects in their arrays. Questions will test the students understanding of Arrays/ArrayLists whether it is developing them, tracing execution, or interacting correctly with them.

Students will trace completed programs from the text. In order to become familiar with the syntax of Arrays/ArrayList in Java and practice mapping out the execution of each, students will type completed programs, adding comments explaining what the code is doing. Students will also be be able to answer a variety of AP Style Multiple Choice Questions pertaining to Arrays/ArrayLists. These questions will involve interacting with/analyzing the impact of different code associated with Arrays/ArrayLists. They will require students to understand concepts of filling/manipulating arrays, separate overall actions of the array with individual elements' actions, test their knowledge of the Arraylists Class, and view two-dimensional arrays as arrays of arrays. In answering these questions students will be required to apply their understanding of these concepts in an analytical manner.

Students will be required to create programs that use Arrays and ArrayList of both primitive and object reference variable types. They will also require interaction with these Arrays/Arraylists. They will have to demonstrate they know the proper structure of Arrays/ArrayLists and how to use indexes to access and interact with specific elements. Students will also show the ability to interact with classes that they have created as well as classes others have created when developing Arrays. Students will use material covered throughout preceding chapters in formulating and interacting with their Arrays. This process will be enhanced through the completion of previous year AP Open Ended Questions from the AP Computer Science Exam. These questions will require the ability to analyze previously developed code and knowledge of how to work with this code and add to it with a focus on processing through Arrays/ArrayLists.

OTHER EVIDENCE

Quizzes/Test
Completion of assigned worksheets/questions
Classroom discussion and interaction during activities
Completion of programming projects using Java Language
Completion of Chapters in AP Review Books (Barron's)
Question of the Day Packet (AP Style Multiple Choice Questions)
Completion of AP Style Open Ended Questions (previous AP Exam test questions)

STAGE THREE

LEARNING PLAN

Java Syntax and Semantics – Arrays/ArrayList (creating and manipulating)

- *Fundamentals of Java:* Chapter 10 &12 Powerpoint
  - Pre-Lab Activies from phschool.com website (pre-test: what do we know?)
  - Tracing/Commenting out of sample programs throughout chapter to explain program's execution
- *Fundamentals of Java:* End of Section and Chapter review questions throughout text in Chapters 10 & 12
- *Java Software Solutions for AP Computer Science A:* end of chapter review questions (Chapter 6)
  - phschool.com website programs: Grading Quizzes, Tracking Sales, Magic Squares, Shopping Cart
- Worksheets form University of Washington Computer Science & Engineering: TEALS Workshop
  - http://homes.cs.washington.edu/~reges/teals/
- Videos that should be viewed in advance of classroom lessons (certain assignments – flip classroom)
  - http://homes.cs.washington.edu/~reges/teals/bjp.shtml
- Online Quizzes and Flash Cards to Monitor Student progress throughout unit
  - http://chortle.ccsu.edu/CS151/cs151java.html

- Online practice questions – codehs.org

Modifying and Creating Interactive Programs with Arrays and ArrayLists

- Students will create classes using JGrasp & BlueJ
- Sample Programming Project:
    - Write a program that initializes an array with ten random integers and then prints four lines of output, containing

        •every element at an even index
        •every even element
        •all elements in reverse order
        •only the first and last element

    - Compute the alternating sum of all elements in an array.
    - Make an array with 10 elements that are multiples of 3. You must use a loop to fill it. No initializer list!
    - Create an array that will store a student's grades. Each time through the loop let the student type in their grade. To determine the size of the array ask the student how many classes they take and use this variable when declaring the array.
        - Modify the grades program so that after the array is filled you calculate the student's average.
        - Modify the grades program so that after the array is filled you determine the lowest and highest grade.
    - Write Java statements for performing the following tasks with an array declared as
      Int [ ] [ ] values = new int [ROWS] [COLUMNS];
        - Fill all entries with 0
        - Fill elements alternately with 0s and 1s in a checkerboard pattern
        - Fill only the elements in the top and bottom rows with zeroes
        - Compute the sum of all elements
        - Print the array in tabular form

- AP Computer Science Open Ended Questions – all previous test questions can be found online
  http://www.skylit.com/beprepared/fr.html
  http://apcentral.collegeboard.com/apc/public/exam/exam_information/2000.html

- Additional Program Possibilities:
  http://homes.cs.washington.edu/~reges/teals/bjp.shtml
  www.phschool.com
  Instructor Resources CD – Fundamental of Java AP Computer Science Essentials (textbook)

Unit Name: Inheritance & Polymorphism
Time Frame: Approximately 8 weeks
Author: Adam Swift

UNIT

Subject: Computer Science | Country: USA

Course/Grade: AP Computer Science (10-12) | State/Group: NJ

School: Egg Harbor Twp High School

UNIT SUMMARY

Students will learn how to take advantage of code re-use better organizing their programs and avoiding redundancy. They will understand class hierarchies including viewing the Object Class as the highest class in the hierarchy. Students will be introduced to what can be inherited and how data and methods are inherited from one class to the next. They will understand that by inheriting variables and methods they will not be required to re-type code and can use the original code as intended. They will be able to identify when inheritance should be used. Students will also understand Polymorphism and how inheritance can allow for polymorphism. They will understand the rules of polymorphism as well as the advantages of implementing it in their programs.

UNIT RESOURCES

Textbook – Fundamentals of Java Chapter 11
Exam View Assessment Suite Chapter 11
Instructor Resources CD Chapter 11 (PPT, Handouts, Solutions)
Textbook – Java Software Solutions AP Computer Science Chapter 7
Website – PreLab Projects, Post Chapter Questions and Projects
AP Computer Science Barron's Test Review Book
College Board AP Computer Science Webpage
Planning Posterboard (mapping programs)
Mini-Whiteboards: Tracing Programs
Software/Compiler – JGrasp & BlueJ
Computers
Pens, Pencils, Markers, & Expo Markers
Internet Access

Internet Resource Links:

http://homes.cs.washington.edu/~reges/teals/
http://homes.cs.washington.edu/~reges/teals/bjp.shtml
http://www.skylit.com/beprepared/fr.html
http://apcentral.collegeboard.com/apc/public/exam/exam_information/2000.html
https://codehs.com/
http://chortle.ccsu.edu/CS151/cs151java.html
http://csunplugged.org/activities/
https://users-mooc.amplify.com/apcs
www.phschool.com

STAGE ONE

GOALS AND STANDARDS

Students will be introduced to Inheritance including how to use it in their programs and being able to analyze its use in completed programs. They will understand the value of inheritance in preventing redundant code. Students will be able to follow class hierarchies and develop interconnected classes. They will be able to created interconnect, succient coding by applying the appropriate rules and processes in using inheritance. Students will also be able to use Polymorphism in their driver programs when interacting with inter-related classes. This unit will improve coding efficiency and demonstrate how creating interconnected classes can improve collaboration. Students will be able to utilize both Inheritance and Polymorphism in their programs and answer multiple choice questions requiring the analyzing of code that uses these concepts.

21st Century.9.CPR2 Apply appropriate academic and technical skills
21st Century.9.CPR6 Demonstrate creativity and innovation.
21st Century.9.CPR11 use technology to enhance productivity
8.2.12.D.3 Determine and use the appropriate resources (e.g., CNC (Computer Numerical Control) equipment, 3D printers, CAD software) in the design, development and creation of a technological product or system.
8.2.12.E.1 Demonstrate an understanding of the problem-solving capacity of computers in our world.
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).
TEC.9-12.9.4.12.K.(4).5 - [Cumulative Progress Indicator] - Use the software development process to design a software application and deliver it to the customer.
ITEC.9-12.9.4.12.K.(4).7 - [Cumulative Progress Indicator] - Implement software testing procedures to ensure quality products.
STEM.9-12.9.4.12.O.(2).6 - [Cumulative Progress Indicator] - Demonstrate the knowledge and technical skills needed to obtain and succeed in a chosen scientific and mathematical field.
TEC.9-12. - [Content Statement] - The design process is a systematic approach to solving problems
TEC.9-12.8.2.12.F.1 - [Cumulative Progress Indicator] - Determine and use the appropriate application of resources in the design, development, and creation of a technological product or system.
12.9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications
12.9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.
12.9.3.IT-PRG.6 Program a computer application using the appropriate programming language

ENDURING UNDERSTANDING

Efficiency and collaboration are essential skills to succeed in the world of Computer Science. The correct implementation of Inheritance will allow for better efficiency. By understanding and applying the concepts of inheritance students will better organize their code and avoid code re-use. In doing so, there code will be easier to read and follow, ultimately facilitating collaboration. Students will come to appreciate the advantages of implementing inheritance and polymorphism when programming. They will also be prepared to collaborate with others when building on their code through their knowledge of how inheritance works.

ESSENTIAL QUESTIONS

Why is efficient programming with limited variables important in the real world?
What is inheritance in programming and how does it support the concept of code re-use?
What is the role of class hierarchies and abstraction when using inheritance while programming?
Why is inheritance useful when creating complex programs?
When using polymorphism how does the compiler know which methods (object reference variables type or objects type) to use?
What is meant by the term open source and how does it apply to programming?
Why is it important to be able to use concepts in our programs as well as analyze/interpret finished code that uses these concepts?
How does an understanding of object oriented programming, regardless of language, prepare you for success in Computer Science?
How is computer science used in today's world and are you capable of correcting, modifying, and creating mini programs?

KNOWLEDGE AND SKILLS

Students will be able to understand, implement, and analyze the concepts of Inheritance and Polymorphism

- Understand class hierarchies (Object Class is at the top of everything)
- Understand how inheritance is implemented in Java
- Identify what can and can not be inherited
- Understand how Inheritance allows for code re-use, improving efficiency
- Be able to create Classes that inherit from others
- Understand the value of overriding methods that are inherited when necessary
- Be able to analyze code created by others that uses inheritance
- Understand the role of the reserved word Super when using inheritance
- Understand what Polymorphism is and when it can be used
- Identify differences between Parent and Child classes
- Understand inheritance/polymorphism as following a trickle down approach

STAGE TWO

PERFORMANCE TASKS

After review of PPT's and classroom dicussions students will answer textbook review questions pertaining to Inheritance and Polymorphism. Students will create visual charts to identify the flow through a hierarchy that follows the rules of inheritance. In implementing inheritance students will reinforce the concept of code re-use and efficiency. Students will also take their hierarchy charts and be able to determine the correct polymorphic references that could be used. Questions will test the students' understanding of Inheritance and Polymorphism whether it is implementing it in their own code or analyzing/following the execution of others' code that uses these concepts.

Students will trace completed programs from the text. In order to become familiar with the syntax and rules of Inheritance and Poloymorphism in Java and practice mapping out the interaction with each, students will type completed programs, adding comments explaining the code's execution. Students will also be be able to answer a variety of AP style Multiple Choice Questions pertaining to Inhertiance and Polymorphism. These questions will involve interacting with/analyzing the impact of different code that implements each of these concepts. These questions will require students to identify whether the concepts are being used correctly. Questions will test the students understanding of the flow of information down the hierarchy (not up). They will also make students identify whether the polymorphic object reference variable is valid and then determine which classes to go to when that reference is used to call methods. In answering these questions students will be required to apply their understanding of Inheritance and Polymorphism in an analytical manner.

Students will be required to create programs that utilize Inheritance and Polymorphism. They will have to demonstrate they know the proper flow of information when using Inheritance as well as what can and can not be inherited. Students will also demonstrate the ability to interact with classes within the same hierarchy. They will also be able to create driver programs that use Polymorphic references correctly based on the inheritance relationships they have developed with their classes. This process will be enhanced through the completion of previous year AP Open Ended Questions from the AP Computer Science Exam. Questions will require the ability to analyze previously developed code and demonstrate the knowledge of how to work with this code and add to it with a focus on Inheritance/Polymorphism (including using the Super Refernce, Overriding Methods, and accessing variables/methods that have been inherited).

OTHER EVIDENCE

Quizzes/Test
Completion of assigned worksheets/questions
Classroom discussion and interaction during activities
Completion of programming projects using Java Language
Completion of Chapters in AP Review Books (Barron's)
Question of the Day Packet (AP Style Multiple Choice Questions)
Completion of AP Style Open Ended Questions (previous AP Exam test questions)

STAGE THREE

LEARNING PLAN

Java Syntax and Semantics – Arrays/ArrayList (creating and manipulating)

- *Fundamentals of Java:* Chapter 11 Powerpoint
    - Pre-Lab Activies from phschool.com website (pre-test: what do we know?)
    - Tracing/Commenting out of sample programs throughout chapter to explain program's execution
- *Fundamentals of Java:* End of Section and Chapter review questions throughout text in Chapter 11
- *Java Software Solutions for AP Computer Science A:* end of chapter review questions (Chapter 7)
    - phschool.com website programs: Exploring Inhertiance, Test Questions, Another Type of Employee
- Worksheets form University of Washington Computer Science & Engineering: TEALS Workshop
    - http://homes.cs.washington.edu/~reges/teals/
- Videos that should be viewed in advance of classroom lessons (certain assignments – flip classroom)
    - http://homes.cs.washington.edu/~reges/teals/bjp.shtml
- Online Quizzes and Flash Cards to Monitor Student progress throughout unit
    - http://chortle.ccsu.edu/CS151/cs151java.html
- Online practice questions – codehs.org

Modifying and Creating Interactive Programs with Inheritance and Polymorphism

- Students will create classes using JGrasp & BlueJ
- Sample Programming Project:

  Write the class SalesPerson which inherits from Employee . Sales people have two additional instance variables: one that represents the value of total sales and one that represents total days traveled. To calculate a SalesPerson's salary you must add commission on their sales to their base salary. If their sales are under 5,000 they get 5% commission, between 5,000 and 10,000 inclusive they get 10%, and over 10,000 they get 15% commission. The SalesPerson's toString method should include its additional instance variables. The SalesPerson class also has a setVacationDays method that takes the person's days traveled as a parameter and modifies vacation days accordingly: for an amount of travel days between 0-10 they get 2 additional vacation days, 10-15 they get 4 additional days, anything over 15 days they get 5 days.

  Complete a driver program that demonstrates that both classes work. Include a polymorphic reference in your program.

- AP Computer Science Open Ended Questions – all previous test questions can be found online
  http://www.skylit.com/beprepared/fr.html
  http://apcentral.collegeboard.com/apc/public/exam/exam_information/2000.html

- Additional Program Possibilities:
  http://homes.cs.washington.edu/~reges/teals/bjp.shtml
  www.phschool.com
  Instructor Resources CD – Fundamental of Java AP Computer Science Essentials (textbook)

Unit Name: Recursion
Time Frame: Approximately 2 weeks
Author: Adam Swift

UNIT

Subject: Computer Science

Country: USA

Course/Grade: AP Computer Science (10-12)

State/Group: NJ

School: Egg Harbor Twp High School

UNIT SUMMARY

With an understanding of efficient programming, students will explore how to create more visually friendly, elegant code. They will understand the advance topic of recursion including how it can be used in place of loops. Students will understand the difficulties in implementing recursion into their code, but gain a respect for the simplicity of code when used.

UNIT RESOURCES

Textbook – Fundamentals of Java Chapter 13
Exam View Assessment Suite Chapter 13
Instructor Resources CD Chapter 13 (PPT, Handouts, Solutions)
Textbook – Java Software Solutions AP Computer Science Chapter 8
Website – PreLab Projects, Post Chapter Questions and Projects
AP Computer Science Barron's Test Review Book
College Board AP Computer Science Webpage
Planning Posterboard (mapping programs)
Mini-Whiteboards: Tracing Programs
Software/Compiler – JGrasp & BlueJ
Computers
Pens, Pencils, Markers, & Expo Markers
Internet Access

Internet Resource Links:

http://homes.cs.washington.edu/~reges/teals/
http://homes.cs.washington.edu/~reges/teals/bjp.shtml
http://www.skylit.com/beprepared/fr.html
http://apcentral.collegeboard.com/apc/public/exam/exam_information/2000.html
https://codehs.com/
http://chortle.ccsu.edu/CS151/cs151java.html
http://csunplugged.org/activities/
https://users-mooc.amplify.com/apcs
www.phschool.com

STAGE ONE

GOALS AND STANDARDS

Students will be introduced to Recursion including how to use it in their programs and analyzing programs that implement it. They will understand the value of recursion in creating elegant, and well *scripted code. Students will be able to follow recursive methods in regards to their execution patterns. They will be able to create recursive code applying the appropriate rules (base case & recursive case) and identifying correct situations to use this technique. This unit will improve code organization and readability. Students will be able to utilize recursion in their programs as well as answer multiple choice questions requiring them to analyze code that includes recursion.*

21st Century.9.CPR2 Apply appropriate academic and technical skills
21st Century.9.CPR6 Demonstrate creativity and innovation.
21st Century.9.CPR11 use technology to enhance productivity
8.2.12.D.3 Determine and use the appropriate resources (e.g., CNC (Computer Numerical Control) equipment, 3D printers, CAD software) in the design, development and creation of a technological product or system.
8.2.12.E.1 Demonstrate an understanding of the problem-solving capacity of computers in our world.
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).
TEC.9-12.9.4.12.K.(4).5 - [Cumulative Progress Indicator] - Use the software development process to design a software application and deliver it to the customer.
ITEC.9-12.9.4.12.K.(4).7 - [Cumulative Progress Indicator] - Implement software testing procedures to ensure quality products.
STEM.9-12.9.4.12.O.(2).6 - [Cumulative Progress Indicator] - Demonstrate the knowledge and technical skills needed to obtain and succeed in a chosen scientific and mathematical field.
TEC.9-12. - [Content Statement] - The design process is a systematic approach to solving problems
TEC.9-12.8.2.12.F.1 - [Cumulative Progress Indicator] - Determine and use the appropriate application of resources in the design, development, and creation of a technological product or system.
12.9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications
12.9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.
12.9.3.IT-PRG.6 Program a computer application using the appropriate programming language

ENDURING UNDERSTANDING

Organization and code readability is essential in the world of Computer Science as collaboration and use of other's code is a fundamental component of it. The correct implementation of recursion will allow for neater and easily followed code. By understanding and applying recursion students will better organize and limit their lines of code. In doing so, there code will be easier to read and follow, ultimately facilitating collaboration.

ESSENTIAL QUESTIONS

Why is efficient programming with limited variables important in the real world?
What is recursion in programming and how does it support efficient code and readability?
What is meant by the term open source and how does it apply to programming?
Why is it important to be able to use recursion in our programs as well as analyze/interpret the code of other programmers' that use it?
How does an understanding of object oriented programming, regardless of language, prepare you for success in Computer Science?
How is computer science used in today's world and are you capable of correcting, modifying, and creating mini programs?

KNOWLEDGE AND SKILLS

Students will be able to understand recursion, implement it, and analyze code that uses it

- Understand recursion and its use in contrast to loops
- Understand how recursion is implemented in Java
- Identify necessary components for recursive methods
- Understand how recursion helps with code efficiency and readability
- Be able to analyze code created by others that uses recursion
- Be able to implement recursion into their own programs

STAGE TWO

PERFORMANCE TASKS

After review of PPT's and classroom dicussions students will answer textbook review questions pertaining to Recursion. Students will create visual charts to identify the execution pattern of methods using recursion. Questions will test the students understanding of Recursion whether it is implementing it in their own code or analyzing/following the execution of others' code that uses the concepts.

Students will trace completed programs from the text. In order to become familiar with the syntax and rules of Recursion in Java and practice mapping out its execution, students will type completed programs, adding comments explaining the code's execution. Students will also be be able to answer a variety of AP style Multiple Choice Questions pertaining to Recursion. These questions will involve interacting with/analyzing the impact of different code implementing this concept. They will require students to identify whether the concept is beings used correctly. In answering these questions students will be required to apply their understanding of concepts in an analytical manner.

Students will be required to create programs that utilize Recursion. They will have to demonstrate they know the proper implementation by including all necessary parts of a recursive method. Students will also show the ability to identify when to use recursion versus a regular loop. Students will be able to create the same method twice (one using a loop and one using recursion) explaining the execution of both.

OTHER EVIDENCE

Quizzes/Test
Completion of assigned worksheets/questions
Classroom discussion and interaction during activities
Completion of programming projects using Java Language
Completion of Chapters in AP Review Books (Barron's)
Question of the Day Packet (AP Style Multiple Choice Questions)
Completion of AP Style Open Ended Questions (previous AP Exam test questions)

STAGE THREE

LEARNING PLAN

Java Syntax and Semantics – Recursion (creating and manipulating)

- *Fundamentals of Java:* Chapter 13 Powerpoint
  - Pre-Lab Activies from phschool.com website (pre-test: what do we know?)
  - Tracing/Commenting out of sample programs throughout chapter to explain program's execution
- *Fundamentals of Java:* End of Section and Chapter review questions throughout text in Chapter 13

- *Java Software Solutions for AP Computer Science A:* end of chapter review questions (Chapter 8)
- Worksheets form University of Washington Computer Science & Engineering: TEALS Workshop
  - http://homes.cs.washington.edu/~reges/teals/
- Videos that should be viewed in advance of classroom lessons (certain assignments – flip classroom)
  - http://homes.cs.washington.edu/~reges/teals/bjp.shtml
- Online Quizzes and Flash Cards to Monitor Student progress throughout unit
  - http://chortle.ccsu.edu/CS151/cs151java.html
- Online practice questions – codehs.org

Modifying and Creating Interactive Programs with Inheritance and Polymorphism

- Students will create classes using JGrasp & BlueJ
- AP Computer Science Multiple Choice Questions using supplemental review books
- Additional Program Possibilities:
  http://homes.cs.washington.edu/~reges/teals/bjp.shtml
  Instructor Resources CD – Fundamental of Java AP Computer Science Essentials (textbook)

Unit Name: AP Exam Review
Time Frame: Approximately 3 weeks
Author: Adam Swift

UNIT

Subject: Computer Science

Country: USA

Course/Grade: AP Computer Science (10-12)

State/Group: NJ

School: Egg Harbor Twp High School

UNIT SUMMARY

This unit will prepare students to take the AP Exam in May. The focus will be on applying concepts learned throughout the last two years to answer AP Style Multiple Choice and Open Ended Questions. Students will further prepare for the expectations for success on the AP Exam. Having learned programming concepts and their application in the real world students will turn their attention to how these concepts will be used to test their knowledge on the AP Exam.

UNIT RESOURCES

Textbook – Fundamentals of Java Chapter
Textbook – Java Software Solutions AP Computer Science
AP Computer Science Barron's Test Review Book
College Board AP Computer Science Webpage
Planning Posterboard (mapping programs)
Software/Compiler – JGrasp & BlueJ
Computers
Pens, Pencils, Markers, & Expo Markers
Internet Access

Internet Resource Links:

http://homes.cs.washington.edu/~reges/teals/
http://homes.cs.washington.edu/~reges/teals/bjp.shtml
http://www.skylit.com/beprepared/fr.html
http://apcentral.collegeboard.com/apc/public/exam/exam_information/2000.html
https://codehs.com/
http://chortle.ccsu.edu/CS151/cs151java.html
https://users-mooc.amplify.com/apcs

STAGE ONE

GOALS AND STANDARDS

Students will focus on preparing for their AP Exam. They will leave this unit comfortable answering the different styles of multiple choice questions that they will encounter on the AP Exam. They will also feel comfortable in anwering AP Style Open Ended Questions by improving their ability to understand what the question is asking and answering these questions in the most succinct manner. Students will have completed multiple AP practice exams learning how to manage their time during the exam.

21st Century.9.CPR2 Apply appropriate academic and technical skills
21st Century.9.CPR6 Demonstrate creativity and innovation.
21st Century.9.CPR11 use technology to enhance productivity
8.2.12.D.3 Determine and use the appropriate resources (e.g., CNC (Computer Numerical Control) equipment, 3D printers, CAD software) in the design, development and creation of a technological product or system.
8.2.12.E.1 Demonstrate an understanding of the problem-solving capacity of computers in our world.
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).
TEC.9-12.9.4.12.K.(4).5 - [Cumulative Progress Indicator] - Use the software development process to design a software application and deliver it to the customer.
ITEC.9-12.9.4.12.K.(4).7 - [Cumulative Progress Indicator] - Implement software testing procedures to ensure quality products.
STEM.9-12.9.4.12.O.(2).6 - [Cumulative Progress Indicator] - Demonstrate the knowledge and technical skills needed to obtain and succeed in a chosen scientific and mathematical field.
TEC.9-12. - [Content Statement] - The design process is a systematic approach to solving problems
TEC.9-12.8.2.12.F.1 - [Cumulative Progress Indicator] - Determine and use the appropriate application of resources in the design, development, and creation of a technological product or system.
12.9.3.IT-PRG.4 Demonstrate the effective use of software development tools to develop software applications
12.9.3.IT-PRG.5 Apply an appropriate software development process to design a software application.
12.9.3.IT-PRG.6 Program a computer application using the appropriate programming language

ENDURING UNDERSTANDING

Computer Science and programming has a broad reach in our society. However, before thinking big picture students must first master the basics required in academia. Much of your career will involve programming and it will be essential to know how to read, interpret, and analyze code. The AP Exam will challenge your skills in this area and preparation for it is essential.

ESSENTIAL QUESTIONS

Why is efficient programming with limited variables important in the real world?
What is the necessary material to know for an introductory college level computer science student?
What is meant by the term open source and how does it apply to programming?
What skills are necessary for success in programming in the real world and in academia?
How does an understanding of object oriented programming, regardless of language, prepare you for success in Computer Science?
How is computer science used in today’s world and are you capable of correcting, modifying, and create mini programs?

KNOWLEDGE AND SKILLS

Students will be able to understand, analyze, interpret, and implement a variety of coding pratices

- Understand basic programming concepts
- Demonstrate the ability to analyze and correct code
- Demonstrate the ability to program – from creating individual programs to building on/using the code of others
- Understand the expections in the world of academia in respect to basic programming knowledge

PERFORMANCE TASKS

After student driven review of material and classroom dicussions students will answer AP Style review questions in preparation for the AP Exam. Students will create visual charts to explain concepts covered throughout the year. Questions will test the students understanding of this material whether it is implementing it in their own code or analyzing/following the execution of others' code.

Students will trace completed programs from a variety of sources. Students will type completed programs, adding comments explaining the code's execution. Students will also be be able to answer a variety of AP style Multiple Choice Questions. These questions will involve interacting with/analyzing the impact of different code. In answering these questions, students will be required to apply their understanding of concepts in an analytical manner.

Students will be required to create and add to programs. They will have to demonstrate they know the proper implementation of different concepts by doing so. The focus will be on AP Exam Open Ended Style Questions. Students will work on solutions as well as evaluating provided solutions and rubrics in preparation for their exam.

OTHER EVIDENCE

Classroom discussion and interaction during activities
Completion of Chapters in AP Review Books (Barron's)
Question of the Day Packet (AP Style Multiple Choice Questions)
Completion of AP Style Open Ended Questions (previous AP Exam test questions)

STAGE THREE

LEARNING PLAN

Java Syntax and Semantics

- *Fundamentals of Java*
  - Tracing/Commenting out of sample programs throughout chapter to explain program's execution
- *Fundamentals of Java:* End of Section and Chapter review questions throughout text
- *Java Software Solutions for AP Computer Science A:* end of chapter review questions
- Worksheets form University of Washington Computer Science & Engineering: TEALS Workshop
  - http://homes.cs.washington.edu/~reges/teals/
- Videos that should be viewed in advance of classroom lessons (certain assignments – flip classroom)
  - http://homes.cs.washington.edu/~reges/teals/bjp.shtml
- Online Quizzes and Flash Cards to Monitor Student progress throughout unit
  - http://chortle.ccsu.edu/CS151/cs151java.html
- Online practice questions – codehs.org

Modifying and Creating Interactive Programs with Inheritance and Polymorphism

- Students will create classes using JGrasp & BlueJ
- AP Computer Science Multiple Choice Questions using supplemental review books
- Additional Program Possibilities:
  http://homes.cs.washington.edu/~reges/teals/bjp.shtml
  Instructor Resources CD – Fundamental of Java AP Computer Science Essentials (textbook)

Curriculum Resources - Differentiated Instruction

Special Education Interventions in General Education

Visual Supports
Extended time to complete tests and assignments
Graphic Organizers
Mnemonic tricks to improve memory
Study guides
Use agenda book for assignments
Provide a posted daily schedule
Use of classroom behavior management system
Use prompts and model directions
Use task analysis to break down activities and lessons into each individual step needed to complete the task
Use concrete examples to teach concepts
Have student repeat/rephrase written directions
Heterogeneous grouping

Resources:

Do to Learn:
http://www.do2learn.com/

Sen Teacher:
http://www.senteacher.org/

Intervention Central:
http://www.interventioncentral.org/

Learning Ally:
https://www.learningally.org/

English Language Learners Interventions in Regular Education

Resources:

FABRIC - Learning Paradigm for ELLs (NJDOE)
www.nj.gov/education/bilingual/pd/**fabric**/**fabric**.pdf

Guide to Teaching ELL Students
http://www.colorincolorado.org/new-teaching-ells

Edutopia - Supporting English Language Learners
https://www.edutopia.org/blog/strategies-and-resources-supporting-ell-todd-finley

Reading Rockets
http://www.readingrockets.org/reading-topics/english-language-learners

Gifted and Talented Interventions in Regular Education

Resources:

Who are Gifted and Talented Students
http://www.npr.org/sections/ed/2015/09/28/443193523/who-are-the-gifted-and-talented-and-what-do-they-need

Hoagies Gifted Education Page
http://www.hoagiesgifted.org/programs.htm

21st Century Learning

Resources:

Partnership for 21st Century Learning
http://www.p21.org/

Career Ready Practices (NJDOE)
http://www.nj.gov/education/cte/hl/CRP.pdf