# AP® Computer Science Principles (APCSP)

**James Madison High School** - Career & Technical Education Course Syllabus and Guidelines- 2016-2017

**Teacher:** Michael S. Soto **Location:** Room H2002

Planning Period: 2<sup>nd</sup> Period (9:42 AM—10:35 AM)

Phone: 210-356-1400 Email: msoto2@neisd.net



## I. Class Overview

This course is recommended for students in Grades 10-12.

AP Computer Science Principles introduces students to the foundational concepts of computer science and challenges them to explore how computing and technology can impact the world. With a unique focus on creative problem solving and real-world applications, AP Computer Science Principles prepares students for college and career.

## II. Textbooks and Materials

**Texts**: Dale, Nell and John Lewis. Computer Science Illuminated. 6th ed. Burlington, MA: Jones & Bartlett Learning, 2016. (CS Illuminated)

Cook, Charles E. Blue Pelican Java. Version 3.0.5K. Refugio, TX: Charles E. Cook: 2010.

Schram, Leon. Exposure Java 2011. Royse City, TX: Leon Schram, 2011.

http://www.schram.org

A pen drive or portable storage device, minimum of (1GB), will be required for saving projects, and a Spiral or Composition notebook dedicated to the course is required, as well.

## **III.** Course Description

Whether it's 3-D animation, engineering, music, app development, medicine, visual design, robotics, or political analysis, computer science is the engine that powers the technology, productivity, and innovation that drive the world. Computer science experience has become an imperative for today's students and the workforce of tomorrow. The AP Program designed AP Computer Science Principles with the goal of creating leaders in computer science fields and attracting and engaging those who are traditionally underrepresented with essential computing tools and multidisciplinary opportunities.

## IV. Software

We will use Snap!, Jeroo, DrJava, JCreator and Java JDK class – all are free and available for download and home use.

## V. Curricular Requirements

- **CR1a** Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P1: Connecting Computing.
- **CR1b** Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P2: Creating Computational Artifacts.
- **CR1c** Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P3: Abstracting.
- **CR1d** Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P4: Analyzing Problems and Artifacts.
- **CR1e** Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P5: Communicating (both orally and written).
- **CR1f** Students are provided with opportunities to meet learning objectives connected to Computational Thinking Practice P6: Collaborating.
- **CR2a** Students are provided with opportunities to meet learning objectives within Big Idea 1: Creativity. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
- **CR2b** Students are provided with opportunities to meet learning objectives within Big Idea 2: Abstraction. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
- CR2c Students are provided with opportunities to meet learning objectives within Big Idea 3: Data and Information. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
- **CR2d** Students are provided with opportunities to meet learning objectives within Big Idea 4: Algorithms. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
- **CR2e** Students are provided with opportunities to meet learning objectives within Big Idea 5: Programming. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
- **CR2f** Students are provided with opportunities to meet learning objectives within Big Idea 6: The Internet. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
- **CR2g** Students are provided with opportunities to meet learning objectives within Big Idea 7: Global Impact. Such opportunities must occur in addition to the AP Computer Science Principles Performance Tasks.
- **CR3** Students are provided the required amount of class time to complete the AP Through-Course Assessment Explore *Impact of Computing Innovations* Performance Task.
- **CR4** Students are provided the required amount of class time to complete the AP Through-Course Assessment Create *Applications from Ideas* Performance Task.

## VI. AP Computer Science Performance Tasks

- Explore Impact of Computing Innovations
- Create Applications from Ideas

Additional Resource: APCSP Exam Reference Guide

During both semesters, emphasis will be placed on building computer science (CS) vocabulary from the APCSP Curriculum Framework, discussion of current topics in CS, and CS theory. Vocabulary building as well as learning objective concepts and essential knowledge statements will be taught through standing class protocols incorporated into a variety of individual and cooperative learning activities.

## VII. Detailed Syllabus

## Unit #1: Semester 1/Week 1

## Big Ideas: Global Impact, Abstraction

#### Text and Resources:

- CS Illuminated: Chapters 1-2
- "Activities," CS Unplugged: Computer Science Without a Computer, CS Education Research Group at the University of Canterbury, NZ (CSU)
- "Binary Game," Cisco, 2011 at cisco.com

#### Content Summary:

#### Introduction to APCSP

- Why is computer science relevant?
- What does computer science involve?
- How does CS impact my area(s) of interest?
- Man vs. Machine (CSU activity)
- Speaking Machine Language (CSU activity)
- Binary Game (Cisco and CSU)
- · Abstraction all around us
- Innovation impact: Three things in a bag
- Vocabulary Unit #1 (Cooperative Learning Strategies)

#### Assessments:

#### **Formative**

- 1. Participation in activities including cooperative learning
- 2. Reflective writing: Innovation
- 3. Cisco game
- 4. CS Illuminated: Chapters 1-2 reading, activities, and discussion

#### Summative

- 1. Unit 1 reflection via Google form response
- 2. Binary counting quiz
- 3. Vocabulary #1 progress check

#### Sample Activities:

- During the first week of class, students participate in activities and discussions to form a basis for the CSP course: What computer science involves (CS), how CS impacts our lives, how CS shapes our innovations and activities, and more. LO 7.2.1[P1] [CR1a] [CR2g]
- We will create a wall of (1) abstraction in everyday life using visual examples and (2) WHY? a list of why CS is relevant to our lives, careers, businesses, governments, etc. By semester two, the question mark "?" of the WHY? will be changed to a "!": WHY! Because at this transition we will see that computer science is not only relevant but also exciting and deserves an exclamation. LO 2.1.1[P3], LO 7.1.1[P4] [CR1c] [CR2f]
- Students will be given a paper sack and asked to bring it back with three items inside that represent their past, present, and future in technology. The items can be actual devices/items or photos of the item. Students will share their innovations as way to introduce themselves and share innovations in computing. LO 1.2.5[P4] [CR1d] [CR2a]

## Unit #2: Semester 1/Week 2

#### **Big Idea: Algorithms**

#### **Text and Resources:**

- CS Illuminated: Chapter 6
- "Google Maps," Google at google.com
- Board games
- Lego NXT robot
- Sphero ball

### **Content Summary:**

## **Computational Thinking**

- Flowcharting
- Pseudocode

#### Programming to Solve a Problem

- Algorithms
  - Write algorithms in natural language
  - ♦ Evaluate algorithms
  - ♦ Think beyond "math" algorithms
- Abstraction
  - Consider how programs use abstractions
  - ♦ Abstractions illustrated by Google Maps
  - ♦ Evaluate abstractions and levels of abstraction in a program
- Lego NXT or Sphero
  - ♦ Program to move in a pattern (e.g., square, figure eight, circle)
  - ♦ Reflect on algorithm(s) used
  - ♦ Reflect on abstraction(s) used

#### Assessments:

#### **Formative**

- 1. Participation
- 2. Flowchart draft
- 3. Algorithm samples
- 4. Abstraction samples
- 5. CS Illuminated: Chapter 6 reading, activities, and discussion

#### **Summative**

- 1. Flowchart final version with pseudocode
- 2. Algorithm quiz
- 3. Programming project video (1 minute)
- 4. Programming code PDF
- 5. Programming flowchart draft
- 6. Vocabulary Unit #2 progress check

#### Sample Activities:

- Students will play board games in groups and create flow charts based on their game movements. Using the flowcharts, students will collaborate to write pseudocode. Board games useful to flowcharting include the following: Sorry, Trouble, Chutes & Ladders, Clue, Uno, Racko, and Mastermind. LO 4.1.2[P5], LO 5.1.3[P6] [CR1e] [CR1f]
- With a partner, students will use software to create digital versions of the flowcharts. The pairs will finish by translating the flowcharts into pseudocode. LO 1.2.2[P2], LO 2.2.1[P2], LO 4.1.2[P5] [CR1b] [CR2a] [CR2b] [CR2d]
- Students work collaboratively to complete a programming lab that solves a given problem. During the semi-guided lab, emphasis will be placed on the algorithm of the block-based code and the abstraction(s) evident. A flowchart, one-minute video, and code PDF will be submitted by each team to demonstrate their program design and functionality. LO 2.2.3 [P3], LO 5.1.2 [P2], LO 5.1.3 [P6], LO 5.2.1 [P3] [CR1c] [CR1b] [CR1f] [CR2b] [CR2e]

## Unit #3: Semester 1/Weeks 3-4

## Big Ideas: Programming, Abstraction

#### Text and Resources:

- "Snap!," University of California, Berkeley at <a href="http://snap.berkeley.edu">http://snap.berkeley.edu</a>
- Windows Moviemaker, iMovie, or similar program

## Content Summary:

- Introduction to block programming
- Days 1-4: Maze Game—Teacher guided and pair programming collaboration
- Days 5-8: Collaborative project enhancement to develop levels, start, and finish to game
- Days 9-10: Project demonstrations and presentations

#### Assessments:

#### **Formative**

- 1. Participation
- 2. Maze project: Submit link
- 3. Algorithm samples from Maze
- 4. Abstraction samples from Maze

#### Summative

- 1. Project video (1 minute)
- 2. Project code PDF with boxes as Performance Task (PT) requires
- 3. Reflection paragraph on collaborative or individual project similar to PT prompts
- 4. Vocabulary Unit #3 progress check (after Week 3)
- 5. Vocabulary test Units 1-3 (end of Week 4)

#### Sample Activity:

• Students will be introduced to block programming in Snap! and will be guided to create a Maze project during the first four days of the unit. Students may work individually or paired with a classmate. After the four-day teacher introduction and overview of the interface, students can work collaboratively or independently to create additional levels to the game. Students may wish to develop a theme for their maze game such as kitten finding yarn or princess finding frog to kiss. The ideas of student decision making and time management will be introduced along with use of prior concepts of flowcharting, use of efficient algorithms, and abstraction development in programming. Students will present their final projects on Days 9-10 of the project. LO 2.2.1[P2], LO 2.2.2[P3], LO 4.2.4[P4], LO 5.1.1[P2], LO 5.1.2[P2], LO 5.3.1[P3] [CR1b] [CR1c] [CR1d] [CR2b] [CR2d] [CR2e]

## Unit #4: Semester 1/Weeks 5-6

## Big Idea: The Internet

#### **Text and Resources:**

- CS Illuminated: Chapter 15
- "Learn," Code Academy at codecademy.com

#### **Content Summary:**

- Introduction to the Internet
  - ♦ History of technology
  - ♦ Timeline of innovation, devices, and usage
  - ♦ Systems of the Internet
  - ♦ URL and IP address scavenger hunt
  - HTML: The language of the web; NOTE: Students will be encouraged to work on Code Academy HTML lessons for supplementary work in coding
  - ♦ Levels of abstraction on the Internet

#### **Assessments:**

#### Formative

- 1. Timeline notes and participation
- 2. Internet systems diagram
- 3. Packet simulation activity
- 4. HTML website activities
- 5. CS Illuminated: Chapter 15 reading, activities, and discussion

#### <u>Summative</u>

- 1. Finished web pages from HTML coding
- 2. Reflective writing from packet simulation activity
- 3. Reflective writing on impact of access to the Internet
- 4. Vocabulary Unit #4 progress check (end of Week 5)
- 5. Internet concepts quiz (end of Week 6)

### Sample Activities:

- A packet simulation whole-class activity will be a two-day interactive activity. Students will have researched the role of various components/devices in packet movement and will simulate the role of each as assigned. Teacher-made "packets" will travel through the classroom "Internet" to see if all are correctly delivered and then assembled into the message as sent. At the end of the activity, students will write a short summary of the simulation, and how well it did or did not demonstrate the characteristics of the Internet. LO 6.2.1[P5], LO 6.2.2[P4] [CR1e] [CR1d] [CR2f]
- Students will be challenged to consider the Internet impact; what does access allow one to do or know when compared to those without access or without skills to utilize the Internet effectively? Students consider socioeconomic impact and global impact of access to technology. LO 7.4.1[P1] [CR1a] [CR2g]

## Unit #5: Semester 1/Weeks 7-8

## Big Ideas: Programming, Abstraction

#### **Text and Resources:**

- CS Illuminated: Chapter 7
- "App Inventor," MIT at appinventor.mit.edu
- "App Inventor Online Book 2," MIT at appinventor.org
- Windows Moviemaker, iMovie, or similar program

#### Content Summary:

- Introduction to Block Programming
  - ♦ Days 1-4: Teacher guided doodle app
  - ♦ Days 5-8: Collaborative projects
  - ♦ Days 9-10: Project demonstrations and presentations

NOTE: Students will be required to submit code printed with red (algorithm) and blue (abstraction) rectangles indicating use of programming components

#### **Assessments:**

#### Formative

- 1. Participation
- 2. Doodle app project: Submit link
- 3. Code PDF with abstractions and algorithms identified
- 4. Explain in paragraph form the algorithm(s) and abstraction(s) in project as PT requires 5. CS Illuminated: Chapter 7 reading, activities, and discussion

#### Summative

- 1. Project video (1 minute)
- 2. Project code PDF with red/blue boxes as PT requires
- 3. Reflection paragraph on collaborative or individual project similar to PT prompts
- 4. Vocabulary Unit #5 progress check (end of Week 7)
- 5. Block programming objective test to include Snap! and AI with code segments

#### Sample Activity:

• Using App Inventor, students will create a Doodle drawing project during the first four days of the unit. After the four-day teacher introduction and overview of the interface, students will independently create another app of their choosing (from App Inventor's online book) or from their own design. The ideas of student decision-making and time management will be discussed along with the use of prior concepts of flowcharting, efficient algorithms, and abstraction development in programming. Students will present their final projects on Days 9-10 of the project. Submission of artifacts will be meant to mimic PT components as a way to scaffold skill set development. LO 5.1.1 [P2] [CR1b] [CR2e]

## Unit #6: Semester 1/Weeks 9-10

## Big Ideas: The Internet, Global Impact, Data and Information

#### **Text and Resources:**

- CS Illuminated: Chapters 16-17
- Caesar cipher websites
- Imitation Game movie: Select clips related to decryption of World War II Nazi messages and use of data obtained

#### **Content Summary:**

- The Impact of The Internet and Data
  - ♦ Cybersecurity
    - O What is at risk?
    - O Who keeps the data gate?
    - O Hardware and software related to security
  - ♦ Cryptography
    - o Caesar cipher
    - Huffman coding activity
    - o Public key encryption
  - ♦ Models and Simulations
    - o Connecting data to how devices and systems collect, feed, and use data
    - O Data models used for decision making
    - Simulations used for predicting and anticipating errors

#### Assessments:

#### **Formative**

- 1. Caesar cipher activity: Create a cipher; decrypt a message from a classmate
- 2. Huffman coding guided activity
- 3. CS Illuminated: Chapters 16-17 reading, activities, and discussion

#### Summative

- 1. Reflective writing on ethical dilemmas related to access to data, security of data, and implications of data misuse
- 2. Vocabulary Unit #6 progress check (end of Week 9)
- 3. Internet concepts test (end of Week 10)
- 4. Vocabulary test, Units 4 and 6 (end of Week 9)

#### Sample Activity:

• Encrypted messages will be created by each student in Caesar cipher format and then decrypted by other students. In addition, examples from World War II and the Enigma machine provide a rich cross-curricular lesson in ethics, innovation, and algorithms. Finally, the Alice and Bob video examples can be used to discuss and analyze public key encryption. LO 6.3.1[P1] [CR1a] [CR2f]

## Unit #7: Semester 1/Weeks 11-12

## Big Ideas: Programming, Abstraction

#### **Text and Resources:**

- CS Illuminated: Chapter 8
- Python.org
- "jGRASP," Auburn University at jgrasp.org
- "Learn," Code Academy at codecademy.com or similar resource

#### Content Summary:

- Introduction to Text Programming
  - ♦ Days 1-4: Teacher guided simple programs with much direction on use of editor
  - ♦ Days 5-10: Student projects
  - ♦ Variables
    - o Types, integers, and real numbers
    - o Self-defining
    - O Using to create algorithms
  - ♦ Operators
    - 0 +, -, /, \*, =
    - o Boolean
    - Conditional statements
  - Abstraction and Algorithm Use
    - Basic calculator project

#### Assessments:

#### **Formative**

- 1. Participation
- 2. Python projects
- 3. Code PDF with abstractions and algorithms identified
- 4. Explain in paragraph form the algorithm(s) and abstraction(s) in project as PT requires
- 5. CS Illuminated: Chapter 8 reading, activities, and discussion

## **Summative**

- Calculator project video (1 minute)
- 2. Calculator project code PDF with red/blue boxes as PT requires
- 3. Vocabulary Unit #7 progress check (end of Week 11)
- 4. Vocabulary test, Units 5 and 7 (end of Week 12)
- 5. Python free response quiz (end of Week 12)

### Sample Activities:

• Students will be introduced to text programming language using Python. Emphasis will be placed on how the code components "look in block vs. look in text." Students will use variables and levels of abstraction to create a basic calculator program using Python. Students will teach each other's calculators for correctness. Submission of artifacts will be meant to mimic PT components as a way to scaffold skill set development. LO 2.2.2[P3], LO 5.3.1[P3], LO 5.4.1[P4], LO 5.5.1[P1] [CR1c] [CR1d] [CR2e]

• Students will work in collaborative teams to create either a checker set of chips or chess set. Depending on the size of the team, students will be assigned a number of game pieces to design based on the group's chosen theme. Various computer-aided design (CAD) programs may be used for this group project based on student selection. Creativity, exploration of innovation, and collaboration are the main areas of focus of this four-week secondary class activity. Student projects will be displayed. LO 1.1.1[P2], LO 1.2.4[P6] [CR1b] [CR2a]

## Unit #8: Semester 1/Weeks 13-14

## Big Ideas: Data and Information, Global Impact

#### **Text and Resources:**

• CS Illuminated: Chapters 3, 18

#### **Content Summary:**

- Data Systems
  - ♦ Collection
  - ♦ Storage
  - Management on devices
  - Management on systems
  - ♦ Management on cloud
  - ♦ Evaluate need for scalability of systems
  - Ethical issues related to data
  - ♦ Legal requirements (HIPAA, FERPA, copyright, Creative Commons)
  - ♦ Impact of data
  - Models and simulations
    - Innovation
    - o Crisis planning
    - O Research and development

#### Assessments:

#### Formative

- 1. Filter, sort, and search a data set
- 2. Cloud use research (student presentations)
- 3. Ethical issue presentations
- 4. CS Illuminated: Chapters 3 and 18 reading, activities, and discussion

#### Summative

- 1. Reflective writing on ethical dilemmas related to access to data, security of data, and implications of data misuse
- 2. Vocabulary Unit #8 progress check (end of Week 9)
- 3. Data unit test including Unit 8 vocabulary (end of Week 10)

#### Sample Activities:

- Students will research and write a report about how data management is impacted by society's need for data (e.g., how is data accessed by devices, where is data collected and used by businesses, what data tells about a person's lifestyle). Students will verify the validity of the information sources and provide the respective citations and references. The report will also include legal issues related to data management. Innovations such as Netflix, Twitter, Instagram, and targeted marketing are topics that will fit well into this assignment. LO 3.1.3[P5], LO 7.2.1[P1], LO 7.3.1[P4], LO 7.5.1[P1], LO 7.5.2 [P5] [CR1e] [CR1a] [CR1d] [CR2c] [CR2g]
- Students will collaboratively filter, sort, and select data from a public data set. LO 3.1.2[P6], LO 3.2.1[P1], LO 3.2.2[P3] [CR16] [CR1c] [CR2c]

## Unit #9: Semester 1/Week 15

## Big Ideas: Data and Information, Global Impact

#### Text and Resources:

• AP Computer Science Principles Performance Task (PT) Explore - Impact of Computing Innovations.

#### Content Summary:

- Mock Explore Performance Task
  - ♦ Possible topics may include:
    - o Google glass
    - o Police body cameras
    - o 3D printer technology
    - o Tesla car
- NOTE: Topics students wish to use for their PT submission should NOT be used for the class Mock PT project.

#### **Assessments:**

#### Formative

- 1. Research about the innovation; discussion and informal presentation
- 2. Participation

#### Summative

- 1. Video artifact as outlined in Explore PT
- 2. Team written submission of one Explore PT prompt as assigned

#### Sample Activity:

• Students will work as a whole-class group and two-to-three person teams to create a Mock Explore PT. A teacher-given topic will be used to allow students to work through the performance task components while experiencing the expectation of the project. Emphasis will be placed on student understanding of the PT prompts and deliverable components. Each student small team will be assigned one prompt to write after whole class research and discussion. Additionally, each team will create a one-minute video about the class topic. The teacher will assemble the written submissions into one final PT document. Using the College Board rubric, the class will work collaboratively to "score" the written document while each team completes a self-assessment on their video artifact. This will give students experience with the rubric language and scoring.

## Unit #10: Semester 1/Weeks 16-17

### Big Ideas: Data and Information, Global Impact

#### Text and Resources:

AP Computer Science Principles Performance Task Explore - Impact of Computing Innovations.

## **Content Summary:**

• Explore PT

#### Assessments:

#### Formative

- 1. Research about the innovation; discussion and informal presentation
- 2. Participation

#### Summative

1. Project deliverables submission to the College Board

#### Sample Activity:

• Students will complete the Explore Performance Task as outlined. Two class weeks or 10 days of 48 minutes each (total of 480 minutes or 8 hours) will be provided in accordance with the College Board project parameters. During these class lab days, the teacher will ensure that students are progressing toward PT completion and that there is understanding of the PT components using the Mock PT experience as a foundation for comparison. [CR3]

## NOTE: Week 18 is Semester Exam week (no new content)

## **End of Semester 1 Syllabus**

## Unit #11: Semester 2/Weeks 1-3

## **Big Idea: Programming**

#### Text and Resources:

- CS Illuminated: Chapter 8
- "Snap!," University of California, Berkeley at snap.berkeley.edu
- Python.org
- "jGRASP," Auburn University at jgrasp.org
- APCSP Exam Reference Guide Content Summary:
- Advanced Programming Concepts
  - ♦ Loops/Iteration
    - o ESPN song in Snap!
    - Revise calculator file from Unit 7 to include a loop feature
  - ♦ Strings
    - Use of strings
  - ♦ Lists
- O Hangman game in Python

#### **Assessments:**

#### **Formative**

- 1. Participation
- 2. Algorithm analysis
- 3. Exam reference guide explanation with block and text code
- 4. Code PDF with abstractions and algorithms identified
- 5. Explain in paragraph form the algorithm(s) and abstraction(s) in project as PT requires
- 6. CS Illuminated: Chapter 8 reading, activities, and discussion

#### Summative

- 1. Project submission
- 2. Free response quiz (end of Week 2)
- 3. Advanced concept programming application test (end of Week 3)

#### Sample Activities:

- Students will revise their own calculator project from Unit 7 to include procedures as well as a loop feature for "play again" yes or no. LO 1.2.3[P2], LO 5.3.1[P3] [CR16] [CR2a] [CR2e]
- Students will use starter code to edit and develop a hangman game in Python. LO 5.2.1[P3], LO 5.4.1[P4], LO 5.5.1[P1] [CR1c] [CR1d] [CR1a] [CR2e]

## Unit #12: Semester 2/Weeks 4-6

### **Big Idea: Programming**

#### Text and Resources:

- CS Illuminated: Chapter 9
- "Java Software," Oracle at oracle.com
- "jGRASP," Auburn University at jgrasp.org
- APCSP Exam Reference Guide

#### Content Summary:

- Java Programming
  - ♦ Introduction to hybrid structure of Java
  - ♦ Introduction to object-oriented programming (OOP)
  - ♦ Classes and methods
  - ♦ Data types
  - ♦ Review of programming concepts in Java syntax
  - ♦ Graphic programming using applets
  - **♦** Inheritance
  - ♦ Modulus

#### Assessments:

#### Formative

- 1. Participation
- 2. Algorithm analysis
- 3. Activity completion
- 4. Code PDF with abstractions and algorithms identified
- 5. Explain in paragraph form the algorithm(s) and abstraction(s) in project as PT requires 6. CS Illuminated: Chapter 9 reading, activities, and discussion

#### Summative

- 1. Project submission
- 2. Java free response quiz (end of Week 5)
- 3. Java programming application test (end of Week 6)
- 4. Programming, algorithm, and abstraction vocabulary test (from Units 3, 5, 7 from Semester 1)

#### Sample Activity:

Java programming will be introduced to students in this three-week unit. Students will complete several guided activities
with the teacher and then will work in pair programming collaborations and individually to complete projects. Graphic
projects will be used to introduce students to the use of methods and parameters in Java and to allow students to create
works of art. Pair programming will be used during teacher-guided instruction and class activities. LO 2.2.2 [P3], LO
5.1.1[P2], LO 5.1.3[P6] [CR1c] [CR1b] [CR2b] [CR2e]

## Unit #13: Semester 2/Week 7

## Big Ideas: Abstraction, Programming

#### **Text and Resources:**

- "Snap!," University of California, Berkeley at snap.berkeley.edu
- Python.org
- "Java Software," Oracle at oracle.com
- "jGRASP," Auburn University at jgrasp.org
- APCSP Exam Reference Guide

#### **Content Summary:**

- Advanced Programming Concepts
  - ♦ Layers of abstraction
  - ♦ Program design
  - ♦ Algorithms
  - ♦ Collaborative troubleshooting
  - ♦ Libraries and APIs

#### **Assessments:**

#### **Formative**

- 1. Participation
- 2. Algorithm analysis
- 3. Abstraction analysis

#### **Summative**

1. Project submission

#### Sample Activity:

• Working with starter code and an API, students will create a complicated program and evaluate the algorithms, abstractions, and design of the program for efficiency and reliability. LO 5.2.1[P3], LO 5.3.1[P3] [CR1c] [CR2e]

## Unit #14: Semester 2/Weeks 8-9

#### Big Ideas: Creativity, Abstraction, Algorithms, Programming

#### **Text and Resources:**

- "Snap!," University of California, Berkeley at snap.berkeley.edu
- Python.org
- "App Inventor," MIT at appinventor.mit.edu
- "Java Software," Oracle at oracle.com
- "jGRASP," Auburn University at jgrasp.org
- Windows Moviemaker, iMovie, or similar program
- AP Computer Science Principles Performance Task Create Applications from Ideas

#### Content Summary:

- Mock Create Performance Task
  - ♦ Possible topics may include:
    - Rock-paper-scissors project
    - Advanced calculator project
    - Hangman game (Snap! or AI)
- NOTE: Project(s) used for the Mock Create PT should NOT be used or enhanced for Create PT individual or collaborative submissions.

#### Assessments:

#### **Formative**

- 1. Participation
- 2. Collaboration reflection

#### Summative

- 1. Mock PT project submission
- 2. Self-assessment of group project
- 3. Individual reflection on group project

#### Sample Activity:

• Students will work in two-to-three person teams to create a Mock Create Performance Task. A teacher given topic will be used to allow students to work through the performance task components while experiencing the expectation of the project. Emphasis will be placed on student understanding of the PT prompts and deliverable components. Student groups may select the programming language for their project from Snap!, Al, Java, or Python. Each student small team will be dividing the written components and deliverables to produce one document for each team. Additionally, each team will create a one-minute video demonstrating execution of their finished project. Using the College Board rubric, the class will work collaboratively to "score" the written documents while each team completes a self-assessment on their video artifact. This will give students experience with the rubric language and scoring.

## Unit #15: Semester 2/Weeks 10-13

## Big Ideas: Creativity, Abstraction, Algorithms, Programming

#### **Text and Resources:**

- "Snap!," University of California, Berkeley at snap.berkeley.edu
- Python.org
- "App Inventor," MIT at appinventor.mit.edu
- "Java Software," Oracle at oracle.com
- "jGRASP," Auburn University at jgrasp.org
- Windows Moviemaker, iMovie, or similar program
- AP Computer Science Principles Performance Task Create Applications from Ideas

#### **Content Summary:**

- Create Performance Task
  - ♦ Weeks 10-11: Collaborative portion
  - ♦ Weeks 12-13: Individual portion

#### Assessments:

#### **Formative**

- 1. Collaborative participation
- 2. Individual completion

### **Summative**

1. Deliverable component submission to the College Board

#### Sample Activity:

• Students will complete the Create Performance Task as outlined. Three class weeks or 15 days of 48 minutes each (for a total of 720 minutes or 12 hours) will be provided in accordance with the College Board project parameters. During these class lab days, the teacher will ensure that students are

progressing toward PT completion and that there is understanding of the PT components using the Mock PT experience as a foundation for comparison. **[CR4]** 

## Unit #16: Semester 2/Weeks 14-16

# Big Ideas: Creativity, Abstraction, Data and Information, Algorithms, Programming, The Internet, Global Impact

#### Text and Resources:

- CS Illuminated: Chapter 18
- Exam Reference Guide for APCSP

#### **Content Summary:**

• Review of AP CSP Concepts, all learning objectives, essential knowledge statements, and vocabulary units

#### **Assessments:**

#### **Formative**

- 1. Review of concepts
- 2. CS Illuminated: Chapter 18 reading, activities, and discussion

### <u>Summati</u>ve

- 1. Vocabulary test on all key terms: Self-test for mastery
- 2. APCSP Practice Exam

#### Sample Activity:

• Class time after completion of the Create PT will focus on student review of course terminology and concepts in preparation for the AP exam. Small group and cooperative learning strategies will be used during the review of course content. The Exam Reference Guide will guide review of programming concepts in block and text syntax. Students will evaluate code samples from projects completed during the course and others provided by the teacher.

NOTE: The APCSP Exam will be in the testing window around Weeks 14-16. Weeks 17-18 (after the AP Exam) will be spent on student individual projects in the language of choice.

## **End of Semester 2 Syllabus and Pacing Guide**

## VIII. Attendance and Course Evaluation

Grades will be based on attendance, participation, and successful completion of various assignments, and projects throughout this course. Quizzes and tests will be given periodically to check your knowledge and multimedia skills

### **Attendance Policy**

An unexcused absence is an absence from school that the parents/guardians are aware of, but was not approved by school administration as an excused absence. Make-up work for unexcused absences will be penalized equal to late work. A 20percent deduction from the total grade earned will be taken on make-up work for unexcused absences.

## **Grading**

Grading Guidelines	
Daily Assignments	25%
Quizzes/Tests/Projects	50%
Final Exam	25%
Total	100%

<b>Grading Scale</b>	
А	90-100%
В	80-89%
С	70-79%
F	<69%

#### **Late Work**

A. Late work is defined as any assignment that is not submitted on the due date and class period with the exception of make-up work for absences or approved school activities.

- B. A 20% deduction from the total grade earned will be taken for late assignments.
- C. Late assignments will be accepted until the material has been assessed summatively or within a three week grading period.
- D. Extenuating circumstances may occur that prevent the completion and turning in of assignments on the due date. It is the parent/guardian and/or student's responsibility to inform the teacher and/or an appropriate administrator of any such circumstances so that an exception to the rule may or may not be granted. The teacher and/or appropriate administrator shall have the authority to render a final decision on the granting of any exceptions.



## **James Madison High School**

Career & Technical Education Department

## Receipt of Pre-AP® Computer Science (Pre-APCS) Syllabus

I have read the Pre-AP® Computer Science (Pre-APCS) Syllabus and understand the expectations and mature behavior that are expected for students who take the course.

Student Name:	
Student Signature:	
Parent's Name:	
Parent's Name:	
Parent Signature:	Date:
Home Phone Number:	
Mom's Cell Phone No.:	
Dad's Cell Phone No.:	
E-mail Address:	
Home Address:	
Zip Code:	