Computer Science 1-3 (AP)
Syllabus/Online Course Plan

Certificated Teacher: ___________________________ Date: 2015-2016

Stage One – Desired Results

Course Title/Grade Level: Computer Science (AP)

Credit: ___one semester (.5) ___X_two semesters (1)

Estimate of hours per week engaged in learning activities
Students will typically commit a minimum of 5 hours of class work per week.

Prerequisites and/or recommended preparation:
Basic Algebra, mathematical representation and logic.

Course Overview: (Include at least one essential question and/or enduring understanding to indicate focus of the course.)
This course is designed to cover the same material present in the first semester of a college level course. In this
course we want to use computers to create new/unique solutions or recycled-new solutions as we explore Object
Oriented Programming (O.O.P.s) to solve problems.

Enduring Understandings for Course:
• design and implement solutions to problems by writing, running, and debugging computer programs.
• use and implement commonly used algorithms and data structures.
• develop and select appropriate algorithms and data structures to solve problems.
• code fluently in an object-oriented paradigm using the programming language Java. Students are expected to be
  familiar with and be able to use standard Java library classes from the AP Java subset.
• read and understand a large program consisting of several classes and interacting objects. Students should be able to
  read and understand a description of the design and development process leading to such a program. (An example of
  such a program is the AP Computer Science Case Study.)
• recognize the ethical and social implications of computer use.

List external resources and include cost for each, (i.e. multimedia resources, 3rd party vendors, subscriptions, etc.)
BPJ_TextBook by Charles E. Cook (pdf) (provided)
JCreator LE (provided)
Current JRE from Sun Systems (provided)
Graphical Labs (BPJ) by Charles E. Cook (pdf) (provided)
AP07_gridworld_Studmanual (pdf) (provided)
Blackboard Account (provided)

Establish Goals: (Grade Level Expectations)
C2—The course includes all of the topics listed in the “Computer Science A” column of the Topic
Outline in the AP Computer Science Course Description.

C3—The course teaches students to design and implement computer based solutions to problems in a
variety of application areas.

C4—The course teaches students to use and implement commonly used algorithms and data structures.

C5—The course teaches students to develop and select appropriate algorithms and data structures to
solve problems.

C6—The course teaches students to code fluently in an object-oriented paradigm using the programming language Java. The course teaches students to use standard Java library classes from the AP Java subset delineated in appendices A and B of the AP Computer Science Course Description. (Note: Students who study a language other than Java in AP Computer Science must also be taught to use Java, as specified in the AP Java subset.)

C7—The course teaches students to read and understand a large program consisting of several classes and interacting objects, and enables students to read and understand the current AP Computer Science Case Study posted on AP Central.

C8—The course teaches students to identify the major hardware and software components of a computer system, their relationship to one another, and the roles of these components within the system.

C9—The course teaches students to recognize the ethical and social Implications of computer use.

<table>
<thead>
<tr>
<th>Understandings:</th>
<th>Essential Questions:</th>
</tr>
</thead>
<tbody>
<tr>
<td>What will students understand (about what big ideas) as a result of the unit? “Students will understand . . .”</td>
<td>What arguable, recurring, and thought provoking questions will guide inquiry and point toward the big ideas of the unit?</td>
</tr>
<tr>
<td><strong>Semester 1</strong></td>
<td></td>
</tr>
<tr>
<td>Unit 1: Basic Java—Students will be introduced to the various primitive data types and distinguishes when to use each.</td>
<td><strong>How can we store data? What can we do with it after it is stored?</strong></td>
</tr>
<tr>
<td>Unit 2: Going with the flow—Students will develop statements and loops to code more efficiently or adapt to circumstance.</td>
<td><strong>How can we write programs that dynamically determine what to do next? How can we accomplish repetitive tasks easily?</strong></td>
</tr>
<tr>
<td>Unit 3: Introducing “Things”—Students will use classes, objects, strings and arrays to develop programs.</td>
<td></td>
</tr>
<tr>
<td>Unit 4: What goes in…—Processing (formatting) inputs from various sources.</td>
<td><strong>How can I make/use Objects to solve problems? How can an array help me to manage large volumes of data? Can an abstract definition make my code more efficient?</strong></td>
</tr>
<tr>
<td></td>
<td>What is a data structure? How can I manipulate inputs so that it may be processed in any way I see fit?</td>
</tr>
<tr>
<td><strong>Semester 2</strong></td>
<td></td>
</tr>
<tr>
<td>Unit 5: Bool Loop— Using GridWorld to review Booleans, switches, and loops.</td>
<td><strong>How can we use Booleans, with loops, to solve problems? Can we combine Booleans and loops to make compact solutions to complicated or arduous tasks?</strong></td>
</tr>
<tr>
<td>Unit 6: Creating an Array of Objects—Using constructors, scanners, and File IO-stream to make an array of objects.</td>
<td>Can we define (overload) a constructor to aid in array implementation? How might an external file be used to initiate or save a program?</td>
</tr>
<tr>
<td>Unit 7: A Review of 2-D Arrays, Inheritance, Exceptions and Recursion.</td>
<td><strong>How can inheritance help objects work together? What is recursion? How might exceptions be used in your program?</strong></td>
</tr>
<tr>
<td>Unit 8: Sorting an Array of Objects —A review of object comparisons to sort, make sets, or write to a file (introduction of data structures).</td>
<td><strong>What methods exist for sorting an array, list, or an arraylist? How can you decide which sort method is better?</strong></td>
</tr>
</tbody>
</table>
**Semester 3 (AP)**

**Unit 9:** A Review of Wrappers, Arrays, and ArrayLists.

**Unit 10:** Data Structures: Linked Lists, Queues, Heaps, and tables.

**Unit 11:** Case Study: A prescribed unit from the College board.

**Unit 12:** self directed project. The student will plan, implement a project of their own design (or one may be assigned).

---

**What is the key knowledge and skill needed to develop the desired understandings?**

What knowledge and skill relates to the content standards on which the unit is focused?

**Students will know:**

How to use the java language to develop algorithms that will solve problems.

**Students will be able to:**

- design and implement solutions to problems by writing, running, and debugging computer programs.
- use and implement commonly used algorithms and data structures.
- develop and select appropriate algorithms and data structures to solve problems.
- code fluently in an object-oriented paradigm using the programming language Java. Students are expected to be familiar with and be able to use standard Java library classes from the AP Java subset.
- read and understand a large program consisting of several classes and interacting objects. Students should be able to read and understand a description of the design and development process leading to such a program. (An example of such a program is the AP Computer Science Case Study.)
- recognize the ethical and social implications of computer use.

---

**Syllabus/Online Course Plan**

**Stage Two – Evidence of Assessment**

What evidence will be collected to determine whether or not the understandings have been developed, the knowledge and skill attained, and the state standards met? [Anchor the work in performance tasks that involve application, supplemented as needed by prompted work, quizzes, observations, etc.]

**Performance Tasks:**

Students will read and practice small pieces of a larger target. The students will then take quizzes and complete projects that may involve several of these pieces. Students will also take Unit Tests which span several quizzes. Throughout this process students will blog about their debugging process and communicate solutions to their peers.

**Other Evidence (self-assessments, observations, work samples, quizzes, tests and so on):**

- Work on a portfolio
- Quizzes through blackboard
- Blogging
- Project submission
## Stage Three – Learning Plan

**Learning Activities:**
What sequence of learning activities and teaching will enable students to perform well at the understandings in Stage 2 and thus display evidence of the desired results in stage one? Possibly use the WHERE acronym to design activities (see below).

Each Unit follows the same sequence.
*Read a section and practice in JCreator LE.*  
*Blog about bugs, errors and other questions.*  
*Complete a review quiz.*  
*Complete a project.*  
*Test.*

Indicate from the table below all applicable learning strategies that may be used in the course.

<table>
<thead>
<tr>
<th>Direct Instruction</th>
<th>Indirect Instruction</th>
<th>Experiential Learning</th>
<th>Independent Study</th>
<th>Interactive Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>x</em>_ Structured</td>
<td><em>x</em>_ Problem-based</td>
<td>__Virt. Field</td>
<td>__Essays</td>
<td><em>x</em>_ Discussion</td>
</tr>
<tr>
<td>Overview</td>
<td><em>x</em>_ Case Studies</td>
<td>Trip</td>
<td>__Self-paced</td>
<td><em>x</em>_ Debates</td>
</tr>
<tr>
<td><em>x</em>_ Mini</td>
<td><em>x</em>_ Inquiry</td>
<td>Experiments</td>
<td>computer</td>
<td><em>x</em>_ Role Playing</td>
</tr>
<tr>
<td>presentation</td>
<td><em>x</em>_ Reflective</td>
<td>__Simulations</td>
<td>__Journals</td>
<td>Panels</td>
</tr>
<tr>
<td>____Drill &amp; Practice</td>
<td><em>x</em>_ Project</td>
<td>__Games</td>
<td>__Learning Logs</td>
<td>____Peer Partner</td>
</tr>
<tr>
<td><em>x</em>_ Demonstrations</td>
<td>____Paper</td>
<td>Field</td>
<td>__Reports</td>
<td>Learning <em>x</em>_ Project</td>
</tr>
<tr>
<td>____Other (List)</td>
<td>____Concept</td>
<td>Observ.</td>
<td>____Directed Study</td>
<td>team</td>
</tr>
<tr>
<td>Mapping</td>
<td>____Other (List)</td>
<td>__Role-playing</td>
<td>__Research</td>
<td>Laboratory Groups</td>
</tr>
<tr>
<td>____Other (List)</td>
<td>Mapping</td>
<td>__Model Bldg.</td>
<td>Projects</td>
<td><em>x</em>_ Think, Pair,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>__Surveys</td>
<td>__Other (List)</td>
<td>Share</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cooperative Learning</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>____Tutorial Groups</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>____Interviewing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>____Conferencing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Other (List)</td>
</tr>
</tbody>
</table>

**Scope and Sequence:**

**Semester One**

**Week 1**
- Read/Notes/Quiz Lesson 1
- Read/Notes/Exercise/Quiz Lesson 2

**Week 2**
- Read/Notes/Exercise/Quiz Lesson 3
- Read/Notes/Exercise/Quiz Lesson 4

**Week 3**
- Read/Notes/Exercise/Quiz Lesson 5
- Project 4-1 “Cheating on your Algebra Assignment”
- Project 5-1 “Mixed Results”
Week 4  <Mid Quarter>
Read/Notes/Exercise/Quiz Lesson 6
Read/Notes/Circle Project/Quiz Lesson 7

Week 5
Read/Notes/Exercise/Quiz Lesson 8
Read/Notes/Exercise/Quiz Lesson 9

Week 6
Read/Notes/Exercise/Quiz Lesson 10
Project 10-1 “Weight on Other Planets”
Test #1 Lessons 1-10. STUDY YOUR NOTES.

Week 7
Read/Notes/Exercise/Quiz Lesson 11

Bottles of Root Beer song…30→1
Read/Notes/Exercise/Quiz Lesson 12

Week 8
Read/Notes/Exercise/Quiz Lesson 13
Read/Notes/Exercise/Quiz Lesson 14

Week 9  END QUARTER
Read/Notes/Exercise/Quiz Lesson 15
Loopy WS  [nested loops]

Week 10
Old McDonald Song Project  [class animal]
Project “Overdrawn at Bank” 15-1
Read/Notes/Exercise/Quiz Lesson 16

Week 11
Read/Notes/Exercise/Quiz Lesson 17
Read/Notes/Exercise/Quiz Lesson 18

Week 12
Encryption Project…
Read/Notes/Exercise/Quiz Lesson 19
Read/Notes/Exercise/Quiz Lesson 20

Week 13
Encryption Project time (See course materials folder)
Test #2 Lessons 11-20 STUDY YOUR NOTES.

Week 14  <Mid Quarter>
Read/Notes /Quiz Lesson 24
Read/Notes/Exercise/Quiz Lesson 25

Week 15
Read/Notes/Exercise/Quiz Lesson 26
Read/Notes/Exercise/Quiz Lesson 27

Week 16
Read/Notes/Exercise/Quiz Lesson 30
Test #3 Lessons 24-30.

Week 17
Project 25-2 Student Averages
Project 30-3 Monte Carlo Technique

Week 18  END QUARTER

Final Game Project (see course materials)

Semester Two

Week 1
Install Bug Jar
Bug Lab 1/journal entry and Bug Lab 2/journal entry

Week 2
Bug Lab 3/journal entry
Bug Lab 4/journal entry

Week 3
Bug Lab 5/journal entry
Bug Lab 6/journal entry

Week 4
Bug Lab 7/journal entry
Bug Lab 8/journal entry

Week 5  <Mid Quarter>
Bug Lab 9/journal entry
Bug Lab 10/journal entry

Week 6
Bug Lab 11/journal entry
Bug Lab 12/journal entry
Bug Lab 13/journal entry

Week 7
Bug Lab 14/journal entry
Bug Lab 15/journal entry

Week 8
Make a bug that has components of each bug lab.
Bug Lab mixed bug/journal entry

Week 9  END QUARTER
Bug Lab 16/journal entry
Bug Lab 17/journal entry

Week 10
Read/Notes/Exercise/Quiz Lesson 33
Bug Lab 18/journal entry
Read/Notes/Exercise/Quiz Lesson 35
Bug Lab 19/journal entry

Week 11
Read/Notes/Exercise/Quiz Lesson 36
Bug Lab 20/journal entry
Bug Lab 21/journal entry

Week 12
Read/Notes/Exercise/Quiz Lesson 37
Read/Notes/Exercise/Quiz Lesson 38
Bug Lab 22/journal entry
Week 13
Read/Notes/Exercise/Quiz Lesson 39
Read/Notes/Exercise/Quiz Lesson 40
Bug Lab 23/journal entry

Week 14
<Mid Quarter>
Unit 4 Test (L:33,35-40) ***One Shot at it***
Read/Notes/Exercise/Quiz Lesson 41

Week 15
Read/Notes/Exercise/Quiz Lesson 42
Read/Notes/Exercise/Quiz Lesson 43

Week 16
Read/Notes/Exercise/Quiz Lesson 44
Bug Lab 24/journal entry
Bug Lab 25/journal entry

Week 17
Read/Notes/Exercise/Quiz Lesson 45
Read/Notes/Exercise/Quiz Lesson 46

Week 18
END QUARTER
Mix Bug, put at least 10 different bug labs together.
Unit 5 Test (L:41-47) ***One Shot at it***

Semester 3 (AP)
Week 1
Retro Bug Lab 23
Read/Notes/Exercise/Quiz Lesson 21

Week 2
Read/Notes/Exercise/ CTB’s /Quiz Lesson 22
Read/Notes/Exercise/ CTB’s /Quiz/Project Milt. Sensor Lesson 23

Week 3
Read/Notes/Exercise/Quiz Lesson 31
Read/Notes/Exercise/Quiz Lesson 32
Bug Lab 27/journal entry
Read/Notes/Exercise/ CTB’s /Quiz/Project PassGravy Lesson 34

Week 4
Read/Notes/Exercise/Quiz Lesson 42
Read/Notes/Exercise/Quiz Lesson 43
Retro Bug Lab 24
Lesson 43 Project “Big Bucks at the Bank”

Week 5
<Mid Quarter>
Read/Notes/Exercise/ CTB’s /Quiz Lesson 44
Retro Bug Lab 25
Read/Notes/Exercise/Quiz Lesson 45

Week 6
Bug Lab 26/journal entry
Read/Notes/Exercise/Quiz Lesson 46
Project (46) HashSet/Intersection

Week 7
Bug Lab 29/ journal entry
Read/Notes/Exercise A&B/Quiz Lesson 47
Bug Lab 30/ journal entry

**Week 8**
Read/Notes/Quiz Lesson 48
Project (48) Optimized Code for Divisors of a Number

**Week 9**
**END QUARTER**
Read/Notes/ CTB’s /Quiz Lesson 49
Read/Notes/ CTB’s /Quiz/Lesson 50

**Week 10**
Project (50) StackLL Class
Project (50) Stack Calc

**Week 11**
Read/Notes/Exercise/Quiz/ProjectReverseOrder Lesson 51
Read/Notes/Quiz/Project BSTfindMethod Lesson52

**Week 12**
Read/Notes/ Exercise/ CTB’s /Quiz/Project Who’sNext Lesson53
Read/Notes/ Exercise/Quiz/Project PrintingHeap/AHeapOfTrouble Lesson 55

**Week 13**
Read/Notes/ Exercise/Quiz/Project Who has PrioLess56

**Week 14**
**<Mid Quarter>**
Read/Notes/ CTB’s /Quiz Lesson 57
Project(56)SmilefortheCamera

**Week 15-17**
Case study

**Week 18**
**END QUARTER**
Make a game that uses, random numbers, and four Sem. 3 ideas.