Huron HS Mathematics (2017-18)
IB Computer Science (APCS-A)
Classroom: Room 4202

Mr. Kevin Behmer
Office: 734.994.2093
behmer@aaps.k12.mi.us

Level: Preparation is for students to take the IBCS Exams test in May of year 2!
Offering: Full Year; Course may count as a Math Elective or 4th year Math course if student has taken and passed Algebra 2.

Prerequisites:

- Successful completion of Geometry.
- A "working knowledge" of computer systems.
- Access to a Java IDE of some type (*BlueJ is our compiler of choice)*
- Prior programming experience very helpful *but not required*!

IB Computer Science (IBCS) is a college-level course designed to cover the materials in a general first-year course at major universities around the world. The programming language used will be Java. Assuming 30 weeks are available prior to the AP exam, the topics covered are listed below. General instructional procedures will include some lecture, programming exercises, projects, and assessments. The time after the AP CS Exam is devoted to a team project of some sort, based upon student interest! NOTE - Guidelines and expectations which are part of the International Baccalaureate (IB) program will also be folded in as available as well especially with regards to the MYP Design Subject Area, Learner Profile characteristics (*http://www.ibo.org/benefits/learner-profile/*), ATL, etc.

Per the IB: Having followed the computer science higher-level course, students will be expected to:

Know and understand:

- relevant facts and concepts
- appropriate methods and techniques
- computer science terminology
- methods of presenting information.

Apply and use:

- relevant facts and concepts
- relevant design methods and techniques
- terminology to communicate effectively
- appropriate communication methods to present information.

Construct, analyse, evaluate and formulate:

- success criteria, solution specifications including task outlines, designs and test plans
- appropriate techniques within a specified solution.

Demonstrate the personal skills of cooperation and perseverance as well as appropriate technical skills for effective problem-solving in developing a specified product.

Additional Local Course Objectives:

- Effectively use critically important procedural programming constructs, including conditionals, looping, and recursion.
- Be able to perform traces on programs, understand pre- and post- conditions and method signatures.
- Learn to code "fluently" in Java in a well-structured fashion and in good style, and properly document code.
- Understand the concept of an algorithm; develop and implement algorithms in Java.
- Analyze/compare efficiency of different algorithms for solving a given problem. Learn to select appropriate algorithms and data structures to solve a given problem.
- Discuss ethical and social issues related to the use of computers

Course Grading:

As this is a college-level course grades are determined with the assessment being the lead categories. Each unit will end with an exam, and periodic quizzes will also be given. This leads to a grade breakdown as follows:

Assessments: 50% Test 20% Quiz
Programming Labs: 15% (code-based)
Programming Ex 10% (completion-based, such as Codingbat, PracticeIt, etc)
Homework 5%
Student work will be evaluated and/or assessed using both traditional evaluation methods as well as IB-based rubrics determined collaboratively by course teachers.

The traditional/district assigned Letter Grades will be used. Ex: A: 93-100 A-:90-92, etc

The Semester grade is also a district mandated weighted Average with Quarter1 and Quarter2 each being 40% and the Semester Exam being 20%.

Students will earn Math Elective credit which may be used for graduation to complete the 4-year requirement as long as that student has also taken and passed Algebra2 or its equivalent!

Internal Assessment - IBCS

The Internal Assessment for IBCS is referred to as 'The Solution.' In essence it is a computational solution to a scenario that is self-selected by the student in collaboration with a client, also of their choosing. The IA comprises 30% of a student's IB grade at the SL level and 20% of their IB grade at the HL Level.

There are 5 components to the IBCS IA and each will be explored and addressed in class prior to the student progressing on their own. They include
Criterion A – Planning
Criterion B – Record of Task & Design
Criterion C – Development
Criterion D – Functionality
Criterion E – Evaluation

The IA is graded locally and included in a student's local grade and then is assessed by the IB as well to calibrate instructor's grading practices!

At present, the intention is to use the IA as the semester exam grade for HL Year2. For students taking CS at the SL level, the IA will be due at the beginning of April prior to testing.

External Assessments:

IB exams are taken in May of Year2. The following papers (exams) are 40%, 20% and 20% respectively of a student's IB score.
Paper 1: Short answer questions covering the entire range of Topics in IBCS
Paper 2: Short answer and code writing covering Object Oriented Program (OOP), which is the IBCS HL core extension.
Paper 3: Essay-based questions regarding a pre-seen case study, published in the May prior to the exam period

Group 4 Project

> As the IB guide indicates, "*The group 4 project is a collaborative activity where students from different group 4 subjects work together on a scientific or technological topic, allowing for concepts and perceptions from across the disciplines to be shared in line with aim 10—that is, to 'encourage an understanding of the relationships between scientific disciplines and the overarching nature of the scientific method*.'"

At Huron the Group4 project will take place in mid-to-late April of students' junior year. The project will be introduced to students in all group 4 classes (Chemistry, Physics, Environmental Systems and Societies, Biology, and Computer Science) starting in February during scheduled time in the school day. Students will submit planning documentation after the initial planning session. Students will have another planning session prior to the hands-on experiment, in which they will be expected to prepare the format of their Group 4 presentation (powerpoint or otherwise), and ready the necessary data collection sheets, time logs, and go over the components of the reflection aspect of the Group 4 Project.

Students will receive a list of suggested topics from which to choose that will incorporate at minimum two sciences, but will be designed to incorporate all sciences offered at Huron High School. Examples of such might involve analyzing soil samples from multiple scientific approaches, or examining water samples for various chemical, biological, physical, and societal properties. The computer science component will be incorporated in a data-modeling capacity, and might involve predictive modeling of pollution levels in water, for example. Student groups will have the additional option to design their own experiments as well. Student groups will need to decide on a topic for their experiment by the end of their first planning session.

The Project will involve hands-on fieldwork for all groups at a local watershed location that will have moving water, flora and fauna, and the opportunity to acquire samples of both. The Project will comprise one day of experimentation and presentation of findings. The morning session will involve collection of data, and the afternoon will be

collaborative time and presentations of the project for grading. The location for the Project will most likely be reached on foot without the need for organized transportation.

Students will be expected to log all hours and document meetings. All findings will be recorded and stored using Google Docs, and shared with all students in the IB sciences at Huron. The school will endeavour to partner with the University of Michigan Environmental Science Department, who will be able to supply some key personnel, materials and resources for measurement purposes on the actual day of experimentation. There are sufficient resources within the science department in the school to support sample collection and analysis of this nature.

As mentioned, students will need to present their collective findings on the Group 4 Project Day, and so will need to prepare the formatting for that presentation as much as possible prior to the actual day in question. Time will be allotted for this planning aspect as well. Students will present as a group, but then producing subject-specific lab work in their respective science classes. Students taking more than one science will need to designate which science they are representing at the outset of the Group 4 planning process in February.

Course Policies:

1. Late programs are accepted up through the following school day only, for 50% credit.
2. Code that does not compile receives zero credit.
3. Code that is not documented receives zero credit.
4. Exams and quizzes that are missed due to excused absence will be made up the day you return to class. Exams/quizzes missed due to unexcused absence are not made up and a zero is entered.
5. Working in partnerships is frequently encouraged. I will designate partner programming many times throughout the year. For these assignments, partners will turn in their code together. HOWEVER, for individual assignments, no two programs should be exactly alike and no solutions copied from the Internet or elsewhere may be submitted. It is very important that you do your own work; there will be nobody there to help you when it comes time for the exam. Academic dishonesty in any form will result in a zero for the assignment as well as possible removal from the course.
6. Computer usage is a privilege. You are expected to abide by the policies in the AAPS computer usage agreement.
7. Attendance every day is a must! The Huron High School attendance policy will be strictly followed in this course.

Students who benefit the most from this course have:

- a general working knowledge of computer systems.
- experience with computer applications.
- an aptitude for computer related topics.
- an aptitude for problem solving.
- a capacity for strong mathematical reasoning.

Materials:

Those students that are most successful typically have:

- binder to hold and organize numerous handouts
- computer access at home w/ internet
- Writing utensils (why you ask, because IB/AP CS exams are pencil and paper tests!)
- Solid ability to work independently and in groups
- Although I do not recommend it, some students opt to use their own computers in class. Keeping one's materials safe and secure is challenging enough when it is just textbooks let alone a computer…

YOU WILL NEED A METHOD FOR KEEPING/TRANSPORTING YOUR FILES – do NOT leave code on the local computers… use Google Drive or a flash drive, school server space, etc. A good rule of thumb is to have multiple backups at any point in time…

Texts and Supplementary Materials:

https://chortle.ccsu.edu/java5/index.html **(Online Textbook)**
Bradley Kjell, Central Connecticut State University

Barron's AP Computer Science, 6th Edition, Roselyn A. Teukolsky. (with CD). (Barron's How to Prepare for the AP Computer Science Advanced Placement Exam)

Absolute Java (used as reference only have several in room) + numerous handouts

Introduction to Programming with Greenfoot (Kolling, 2009)

Blue Pelican Java, (Cook, Charles E., Virtualbookworm Publishing, 2013)

Please download this pdf file (free) for your reference. We will use lessons from this book. www.bluepelicanjava.com

Downey, Allen, *How To Think Like A Computer Scientist (Java),* Eleven Learning, 2011. Please download this pdf file (free) for your reference. www.greenteapress.com/thinkapjava/

Students should establish an account at:

- *Khan Academy*
- *codingbat.com:* http://codingbat.com/java.
- *Practice It!* http://practiceit.cs.washington.edu/

Download & Install our Compiler, BlueJ (Java IDE), at home: www.bluej.org

Ref. manual: www.bluej.org/doc/bluej-ref-manual.pdf

You will also use current media sources and Internet articles and blogs discussing technological, ethical and social issues related to computer use.

Course Outline

All time frames are approximate as this course is in perpetual update mode! Assignments, materials, etc will be updated/posted on a rolling schedule!

**********1st Semester**********

1. Unit 1: Computer Basics (2 weeks)
2. Unit 2: Java Basics (4.5 weeks)
3. Unit 3: Object Oriented Programming (OOP) Intro (5.5 weeks)
4. Unit 4: Resource Management (2 weeks)
5. Unit 4: Computational Thinking & Program Development (2 weeks)

**********2nd Semester**********

6. Unit 6: System Design Basics
7. Unit 7: Systems in Organizations
8. Unit 8: Abstract Data Types
9. Unit 9: Computer Organization
10. Unit 4: Case Study & Internal Assessment Introduction

Unit 2: Procedural "Boot Camp": Arithmetic, logic, & control statements (8 weeks)

1. Data types, variables, and arithmetic (duration 1 week)

The concepts of a variable and a data type. Declarations of variables. Fields vs. local variables. The primitive data types: int, double and char. Literal and symbolic constants. Initialization of variables. Scope of variables. Arithmetic expressions. Data types in arithmetic expressions. The cast operator. The compound assignment (+=, etc.) and increment and decrement operators (++, --). Converting numbers and objects into strings.

Reading and exercises: Online book: ch 8-11. BPJ Lessons 1-5.
Lab: Making Change, Distance Formula, MET calorie counter
Lab: Codingbat exercises Warmup-1, String-1
Lab: Coding Set 1

2. Strings (duration 1 week)

String objects. Literal strings. Immutability. String methods. Converting strings into numbers and numbers into strings. The Character class and its methods.

Reading and exercises: BPJ Lesson 3 and 17. Selected exercises
Lab: Palindromes, Two-part numbers revisited
Extra: codingbat.com *String-1*, *String-2*, *String-3*.

3. Conditionals (duration 3 weeks)

The if-else statement, Boolean expressions, the boolean data type, true and false values. Relational and logical operators. De Morgan's laws. Short-circuit evaluation. Nested if-else and if-else-if. The switch statement. The Math class. Introduction to the Finch Robot.

Reading and exercises: Online book: ch 12-14. BPJ Lesson 6-10, 31 with selected exercises.
Lab: boolean expressions WS, Coding Set 2
Lab: AP Chatbot lab
Extra: Codingbat Logic-1 and *Logic-2*.

4. Loops/Iterative statements and more on methods (duration 3 weeks)

while, for, and do-while loops. break and return in loops. Parameters, return types. The concept of an algorithm. Flow charts, pseudocode. Arrays.

Reading and exercises: Online book: ch 14-20. BPJ Lesson 11-13 with exercises. BPJ lesson 18.
Lab: Codingbat: String-2, Practice-It exercises
Lab: Looping Exploration Lab
Lab: Coding Set 3
Lab: Using Arrays. Codingbat: Arrays-1, Arrays-2
Project: Finch Software Project *Finch Plays Simon*

Unit 3: Intro to OOP (7 weeks)

1. Details of defining classes and using objects (duration 3.5 weeks)

Public and private fields and methods. Constructors and the new operator. References to objects. Calling methods and accessing fields. Passing parameters and objects to constructors and methods. return statement. Overloaded methods. Overriding methods. Static variables and methods. Intro to Inheritance. Objects as data types. Encapsulation.

Reading and exercises: Online book: ch 25-35. BPJ Lesson 14, 15, 20, and *Introduction to Greenfoot* (Kolling)
Lab: Coding Set 4 (Fraction class, Coin class, PigLatin etc.)
Lab: AP practice (writing classes FRQ)

2. Greenfoot (duration 3.5 weeks)

Greenfoot textbook is used for this section – section concludes with the assignment of the "big Greenfoot project". Also review for final exam.

Lab: Fatcat
Lab: Working in Greenfoot- Little Crab Scenario (ch 1-4)
Lab: Working in Greenfoot - creating an Interactive Piano

**********1st Semester Exam**********

**********2nd Semester Exam**********

Unit 4: Inheritance (4 weeks)

1. Class hierarchies, abstract classes, and interfaces (duration 2.5 weeks)

Class hierarchies and is-a/has-a relationships. Abstract classes. Invoking superclass's constructors and calling superclass's methods. Polymorphism. Interfaces. Exceptions.

Reading and exercises: Online book: ch 50-54. BPJ Lesson 36 – 37, 44, selected exercises. Past AP free-response questions on class hierarchies and polymorphism.
Lab: Abtract class lab
Lab: Fractions with the Comparable interface

2. Two-Dimensional Arrays (duration 1.5 weeks)

Review one-dimensional arrays. Declaring and initializing. Indices. Length. IndexOutOfBoundsException. Two-dimensional arrays. Accessing the number of rows and columns. Traversals and the "for-each" loop. Inserting and removing elements.

Reading and exercises: Online book: ch 46-49. BPJ Lesson 18-19, 33-34,
Lab: AP Picture Lab
Lab: Deadly Desert or *Soduko*
Extra: codingbat *Arrays-1*, *Arrays-2*, *Arrays-3*.

Unit 5: The List interface, the ArrayList class, searching and sorting , recursion (5 weeks)

1. ArrayLists (duration 1 week)

ArrayList's structure. The List interface. ArrayList's constructors and methods. Pitfalls.

Reading and exercises: BPJ Lesson 41, 42
Lab: Creating an Index for a Document
Lab: Past AP free-response questions on ArrayList.
Lab: Coding set 5.

2. Searching and sorting. (duration 2 weeks)

Comparing objects. Reviewing equals method and the Comparable interface. Sequential and Binary Search. Selection Sort, Insertion Sort, and Mergesort. The number of comparisons required in Sequential and Binary Search. Comparison of efficiency, Big O notation.

Reading and exercises: online book ch. 90-91. BPJ Lesson 50, 40, 38 with selected exercises.
Lab: Unplugged sorting lab
Lab: codingbat sorting problems
Lab: Comparing efficiency of several sorting algorithms.
online Lab: Graphically comparing sorting algorithms.

3. Recursion (duration 2 weeks)

Iterations. Recursion. Introduction to efficiency. Preconditions and Postconditions. Fibonacci and Factorial – comparing recursion with iterative method.

Reading and exercises: Online book ch. 70-75. BPJ Lesson 39, 47, and supplementary articles.
Lab: Review exercises with Arrays. Codingbat: *Arrays-3, Diving Judges problem*.
Lab: coding set 6.
Lab: AP Curriculum Module: Recursion (selected sections)

Unit 6: AP Exam Review & Social Implications of Computing (3 weeks)

Social implications of computing (duration TBD)

Reading: NSA article, "No expectation of privacy" article (Google), data mining, metadata, ethics of computing, software and music piracy, issues w/r/t artificial intelligence, internet censorship, etc. I have several other articles as well.
Project: Critically examining the effects of a technology innovation.

Unit 7: Enrichment Unit
(optional, duration varies, depends on timing w/r/t AP Exam)

1. Streams and files

Text and binary files. Streams vs. random-access files. Java I/O package. The Scanner class. Checked exceptions.

Reading and exercises: BPJ Lesson 23-25.

2. Graphics and GUIs

Computer graphics concepts. The Java Graphics class. GUI components and their events. Layouts. Handling mouse and keyboard events.

Reading and exercises: TBD

3. Lego Mindstorms in Java
LeJOS

Students will construct and program a robot to compete against other robots. All programming will be done in Java using the LeJOS libraries. Programs must be written with proper style and properly documented. This project will count as the 2nd semester final exam grade.

NOTE: While experiencing the above topics, you will find that problem solving, algorithm development, team building, and computational thinking are woven throughout the course. Computer Science will require all of the skills you have learned in other courses, including math and English language skills, as well as applied sciences. In support of the AAPS district reading and writing goals, we will work to improve skills in these areas as well so as to cultivate communication skills that are highly desirable in today's business market!

Ann Arbor Huron Career & Technical Education (CTE) Mission Statement

The Huron High School Career and Technical Education (CTE) department promotes equity and access for all Ann Arbor Public Schools students to CTE courses. Classes are inquiry-based, hands-on, and incorporate global awareness through real-world problem solving activities. The CTE department prepares students for employment and/or higher education through collaboration with local business and industry partners. Leadership and community service opportunities are available through career and technical student organizations such as BPA, DECA, FCCLA, HOSA, and SkillsUSA.

COMPUTER LAB/CLASSROOM RULES & REGULATIONS

1. Students are expected to follow the rules stated in the Ann Arbor Public Schools Acceptable Use Policy for computer usage.

2. ABSOLUTELY NO GUM, CANDY, ETC. WILL BE PERMITTED IN THE COMPUTER LAB. WATER is allowed as long as it is kept in a secure container on the floor at your work station.

3. Students will not attempt to access personal data or email of others, including peers, teachers and administrators. Students will not share passwords to their files or try to break desktop or network security of any machine in the building. Students will not use lab computers for personal activities beyond the requirements of the coursework.

4. Students should promptly report any problems encountered regarding district electronic equipment.

5. Students who fail to adhere to class rules and regulations will earn a blue lunch detention and further action if necessary.

6. Follow all teacher directions and observe classroom norms: School-appropriate language only, NO CELL PHONE USE during class (except when directed), treat fellow students with respect. Remaining quiet and engaged when someone (teacher or otherwise) is speaking. This class is a semi-professional environment and you are expected to conduct yourself accordingly.

7. Code submissions that are identical will be graded as ZEROS. Collaboration and assistance are different than cheating. It's fine to point someone in the right direction, but it's wrong not to allow him/her to make the trip. If someone is asking you to give up your solution or work please let me know immediately.