

**Jefferson County High School  
Course Syllabus**

**A. Course Programming & Logic I (Pre-AP Computer Science)**

**B. Department Career & Technical Education**

**C. Course Description** Programming and Logic is a course in which students will develop skills in problem analysis, construction of algorithms, and computer implementation of algorithms as they work on programming projects of increasing complexity. Students will begin learning the Java programming language and apply to create computer applications. Emphasis is on actual programming projects, both individual and group. Course content will be repeatedly applied to increasingly complex projects. This course will prepare students for Programming and Logic II which includes the AP Computer Science Exam. It is strongly recommended that students have access to a computer outside of class. All necessary software will be provided by the instructor. Students are strongly encouraged to take Programming and Logic I & II in the same school year

**D. Grade Term Semester, however students are strongly encouraged to take P&L I & II in the same school year and attempt to earn AP credit in P&L II.**

**E. Grading Scale**

<u>Range</u>	<u>Honors/ Regular</u>	<u>College-Level</u>	<u>A.P.</u>
93-100 A	4.0	4.5	5.0
85-92 B	3.0	3.5	4.0
75-84 C	2.0	2.5	3.0
70-74 D	1.0	1.5	2.0

**F. Term Dates**

- a. 1<sup>st</sup> 9 Weeks August 5, 2016 – October 7, 2016
- b. 2<sup>nd</sup> 9 Weeks October 8, 2016 – December 16, 2016
- c. 3<sup>rd</sup> 9 Weeks January 5, 2017 – March 15, 2017
- d. 4<sup>th</sup> 9 Weeks March 16, 2017 – May 25, 2017

**G. Textbook(s)** *A variety of current online resources will be used.*

**H. Other Required Reading - N/A**

**I. Other Resources**

- a. Odysseyware
- b. Moodle
- c. Office 365

- d. PluralSite.com
- e. CodingBat.com
- f. CodeCombat.com
- g. CodeHS.com

#### **J. Major Assignments**

- a. *Students will create a number of functional programs to complete a variety of tasks.*

#### **K. Procedures for Parental Access to Instructional Materials**

- a. Aspen Parent Portal
- b. Instructor's Website
- c. Email Instructor
- d. Parent Teacher Conference
  - a. There are two designated conference dates during the school year. Parents who would like to request additional meetings may make appointments for conferences with the teachers (during their planning periods), counselors, or a principal by telephoning the school office.

#### **L. Field Trips**

- a. Any schedule fieldtrip will have a definite educational purpose and will reflect careful planning. Signed permission forms will be obtained when an off campus trip is planned.

#### **M. Standards & Objectives**

##### **Computer Science Overview**

I can use news articles and instructional materials, investigate key milestones in the development of computers and logical devises. (1<sup>st</sup> Nine Weeks)

I can create and present a document and/or illustration depicting the timeline of development that led to modern-day operating systems, programmable controllers, and widespread digital communications via the Internet and wireless networks, citing specific textual evidence. (1<sup>st</sup> Nine Weeks)

I can compare and contrast the benefits, features, and typical applications of common modern programming languages and environments. (1<sup>st</sup> Nine Weeks)

I can craft an argument to defend the choice of a certain language to solve a particular problem, developing claim(s) and counterclaim(s) with specific textual evidence and reasoning. (Ongoing)

##### **Ethics**

I can use news articles and text of legislation, analyze ethical programming practices, including but not limited to the issues of confidentiality, privacy, piracy, fraud and misuse, liability, copyright, open source software, trade secrets, and sabotage. (1<sup>st</sup> Nine Weeks)

## **Programming Skills**

I can differentiate between system-level and application solutions, and identify an appropriate code-based strategy to solve a given problem. (Ongoing)

I can apply the system management tools present in a programming development environment to:

- a) Select the most appropriate programming language for the task at hand (Ongoing)
- b) Develop syntactically correct program code using current best practices and emerging classes of development techniques (Ongoing)
- c) Use a compiler to interpret the source code and produce executable program code (Ongoing)

I can develop strategies that work within the constraints of major operating system fundamentals, in the process of developing and implementing programming solutions, such as:

- a) Security protocols and procedures for accessing files and folders (Ongoing)
- b) File management syntax requirements, including but not limited to creating, naming, organizing, copying, moving, and deleting files (Ongoing)
- c) File naming conventions, as they apply across multiple software applications and file types. (Ongoing)

I can write pseudocode and construct a flowchart for a process before starting to develop the program code. (Ongoing)

I can organize and develop a plan to acquire and manage the data values for a process, including the following:

- a) Data types, such as string, numeric, character, integer, and date (Ongoing)
- b) Program variable names (Ongoing)
- c) Variables and constants (Ongoing)
- d) Arrays (at least one- and two-dimensional), subscripts (2<sup>nd</sup> Nine Weeks)
- e) Input from files and user responses (Ongoing)
- f) Output to files and reports (Ongoing)

I can use a programming language specified by the instructor, convert the pseudocode for a selected process to program code, incorporating at least three of the following structures, the need for which will be dictated by the assigned problem(s) and process(es). The resulting code design can be event-driven, object-oriented, or procedural.

- a) Operations and functions (user-defined and/or library) (Ongoing)
- b) Repetition (loops) (Ongoing)
- c) Decision (if...else, case) (Ongoing)
- d) Recursion (2<sup>nd</sup> Nine Weeks)

I can verify the correct operation of the resulting program code with several test cases:

- a) All valid values (Ongoing)

- b) Error trapping of invalid values (Ongoing)
- c) Error trapping of invalid program operation (Ongoing)
- d) Troubleshooting/remediating program problems (Ongoing)

### **Project Planning and Quality Assurance**

I can compile the necessary documentation to understand the nature of a computer programming problem and the customer/client specifications for the request and summarize in an informational text. This will include evidence of the scope of the problem, its attendant input and output information, the required system processing, and the software specifications involved. (Ongoing)

I can analyze a given problem and develop a coherent strategy in the form of a project plan to meet the customer/client's need. The plan will include, but will not be limited to, defining the project scope as addressed by the problem documentation, identifying software development and implementation issues, timeline and benchmarks for design, and addressing issues associated with software maintenance and life cycle. (Ongoing)

I can articulate the nature of the program designs by creating documentation, in the software development process, that addresses topics including but not limited to:

- a) The procedural, object-oriented, event-driven, or other nature of the various portions of the resulting application (Ongoing)
- b) The data structures used for inputs, outputs, and internal manipulations (Ongoing)
- c) The algorithms and guiding formulas used (Ongoing)
- d) Constraints on accurate operation and results (Ongoing)
- e) Modular designs that enable portability (Ongoing)
- f) Interface details that permit ready maintenance and upkeep (Ongoing)

I can apply principles of quality assurance during application development to certify bug tracking, audit trails, testing results, and other quality considerations. (Ongoing)

I can annotate each quality assurance task with evidence from best practices endorsed by industry or research. (Ongoing)

I can document the security risks associated with new applications and evaluate the severity of the risk involved in each, including but not limited to:

- a) Identifying threats to information systems facilities, data communications systems, and other applications (2<sup>nd</sup> Nine Weeks)
- b) Adhering to federal and state legislation pertaining to computer crime, fraud, and abuse (2<sup>nd</sup> Nine Weeks)
- c) Providing means for preserving confidentiality and encryption of sensitive data (2<sup>nd</sup> Nine Weeks)
- d) Detailing steps to recover from routine errors or catastrophic failures, such as might be caused by a malicious computer virus. (2<sup>nd</sup> Nine Weeks)