Course Title – Computer Programming II

Implement start year – 2015-2016

Revision Committee Members, email, extension –

Scot Butler	sbutler@lrhsd.org	**ext. 8870**
Chris Callinan	ccallinan@lrhsd.org	**ext. 8364**
Robert Kibler	rkibler@lrhsd.org	**ext. 8583**

Unit # 1 – Introduction to Algorithms

Transfer Goal –

Students will be able to independently use their learning to model an algorithm to complete a specific task.

Stage 1 – Desired Results

Established Goals

2009 NJCCC Standard(s), Strand(s)/CPI #
(http://www.nj.gov/education/cccs/2009/final.htm)

Common Core Curriculum Standards for Math and English
(http://www.corestandards.org/)

9.4.12.K.66 Employ information management techniques and strategies to assist in decision-making

9.4.12.K.(3).8 Participate in a user-friendly design and development process Web-based and digital communication solution

9.4..12.K.(4).1 Identify and analyze customer software needs and requirements to guide programming and software development

9.4.12.K.(4).2 Create and use information technology strategies and projects plans when solving specific problems to deliver a product that meets customer specifications

9.4.12.K.(4).3 Identify and analyze system and software requirements to ensure maximum operating efficiency

21st Century Themes

(www.21stcenturyskills.org)

_x_ Global Awareness
___Financial, Economic, Business and Entrepreneurial Literacy
___Civic Literacy
___Health Literacy
___Environmental Literacy

21st Century Skills

Learning and Innovation Skills:
_x_Creativity and Innovation
_x_Critical Thinking and Problem Solving
___Communication and Collaboration

Information, Media and Technology Skills:
_x_Information Literacy
___Media Literacy
_x_ICT (Information, Communications and

9.4.12.K.(4).4 Demonstrate the effective use software development tools to develop software applications 9.4.12.K.(4).5 Use the software development process to design a software and deliver it to the customer 9.4.12.K.(4).6 Produce a computer application, in code, to demonstrate proficiency in developing an application using the appropriate programming language	Technology) Literacy *Life and Career Skills:* _x_Flexibility and Adaptability ___Initiative and Self-Direction ___Social and Cross-Cultural Skills _x_Productivity and Accountability ___Leadership and Responsibility
Enduring Understandings: *Students will understand that . . .* *EU 1* careful design of programs based on the specified requirements is necessary prior to beginning to write the code *EU 2* pseudo code and flowcharts can model a program and ensure that it meets the specified requirements *EU 3* programming languages have evolved from low-level to high-level languages and this evolution is an ongoing process *EU 4* algorithms increase the efficiency of the coding of a program *EU 5* computers store and read data using various systems	**Essential Questions:** *EU 1* • Why is it important to plan a program prior to coding? *EU 2* • How is a flow chart developed, read and interpreted? • How are flowcharts used to develop pseudo code? • Why are flowcharts and pseudo code important? *EU 3* • How has programming developed from its beginning stages to the current state? • Why is programming constantly evolving? *EU 4* • What is the purpose of an algorithm? • How do algorithms aid in coding? *EU 5* • How does a computer read and store data? • How do numbers translate to various number systems?

<u>Knowledge:</u> *Students will know . . .*	**<u>Skills:</u>** *Students will be able to . . .*
EU 1 • the process in planning a program • the steps to ensure that the specified requirements are met	EU 1 • create a program plan given a specific situation • check to see the specified requirements are met
EU2 • the method to design, create and use a flowchart • the process to translate flowcharts into pseudo code • the importance of careful development of flowcharts and pseudo code prior to coding	EU 2 • develop a flowchart from a plan • translate a flowchart into pseudo code • explain why proper planning prior to coding is an important step in coding
EU 3 • the difference between machine and high level languages • why programming languages are continually evolving to adapt to the needs of society	EU 3 • compare and contrast machine and high-level languages • explain how and why programming languages are evolving
EU 4 • the definition of an algorithm • the procedures used to develop algorithms to increase the efficiency of the coding of a program	EU 4 • identify key aspects of a problem posed and problem solve • modify and adapt algorithms to aid in the development of code
EU 5 • Binary, Octal, Decimal and Hexadecimal are the number systems used by computers	EU 5 • convert numbers between all of the computer number systems

Stage 2 – Assessment Evidence

Recommended Performance Tasks:

Other Recommended Evidence:

- Flow charts including program flow, evaluations of conditions and the looping process
- Pseudo code: a handwritten version of the program where the code is not written in full, but rather the process that a program will follow that is developed on paper
- Algorithms: Written code of the mathematical process that will allow data to be put into an array, sorted and then have information be retrieved from using different techniques
- Quizzes on software, number systems, and program planning
- Unit test
- Class discussion

Stage 3 – Learning Plan

Suggested Learning Activities to Include Differentiated Instruction and Interdisciplinary Connections:

- Research early programming languages and the evolution process in the modification of these languages (A)
- Convert numbers for any of the number systems to the other three systems (M)
- Develop an algorithm for walking from the classroom to the cafeteria starting with standing up from the desk and ending with sitting at a table in the cafeteria (T)
- Develop an algorithm for converting a temperture from Fahrenheit to Celsius and vice versa (T)
- Convert algorithms into a flowchart (T)
- Desk check the algorithm for converting temperature (T)
- Debug an existing algorithm and convert into a flowchart (T)

The following is the suggested sequence of learning activities and number of days:
(Approximately 27 days)

- **Evolution of computer system: History, Language Levels (Lower and Higher)**
- **Problem Solving Techniques: Problem Definition, Top Down Design**
- **Problem Solving Techniques: Algorithms and Planning**
- **Problem Solving Techniques: Flowcharting, Documentation and Debugging**
- **Program Design: Working with the user interface**
- **Organization of Program Outcomes: Working with interfaces**
- **Program Outputs: List Boxes, Consoles and output interfaces**

Vocabulary

- Programming Languages
- Machine Languages
- Bit/Byte
- Object-Oriented Design
- User interface
- Variable
- Documentation/Comments
- Code
- Binary
- Octal
- Hexadecimal
- Pseudo code
- Flow chart
- Algorithm