

## **COURSE PROFICIENCY OUTLINE**

### **Introduction to Computer Programming - 340**

5 Credits

Purpose

Introduction to Computer Programming is an introductory course in computer programming designed for the student who has successfully completed Algebra I Honors. It is highly recommended that any student wishing to major in Computer Science, Math, Engineering, or Science take this course since it will most likely be a requirement in college. This course serves as a prerequisite to AP Computer Science and Topics of Modern Programming.

In Introduction to Computer Programming, the student will develop computer science skills of algorithm development, problem solving and programming using software engineering principles. Students will design algorithms and programming solutions to a variety of computational problems using the C++ programming language as its tool.

Programming will give the student a sense of what computers can and cannot do, and is one of the best ways to gain a deep understanding of computer technology. Learning to program gives an appreciation for the important and difficult task of software creation. Programming will give the student problem solving skills that can be used in any occupation.

#### I. Student Outcomes                      4.2, 4.3, 4.4, 4.5

- A. The student will become familiar with the microcomputer environment.
- B. The student will recognize the three basic elements in all computer programs: Input, Process, and Output.
- C. The student will practice algorithmic thinking by breaking down problems into smaller tasks and solving problems in a step by step manner.
- D. The student will learn about principles, methodologies and applications of computer science in the modern world.
- E. The student will be taught a general knowledge of computer hardware, software, languages and their impact in the modern world.
- F. The student will learn about the legal and ethical behaviors when using information and technology, and discuss consequences of misuse.
- G. The student will be exposed to different careers in computing.
- H. The student will write programs that exhibit style, modularity, and documentation.
- I. The student will learn about the connection between elements of math and computer science including binary numbers and logic design.
- J. The student will develop an understanding of the three prime programming structures: Sequential, Branching, and Looping.
- K. The student will be able to develop program logic and translate the logic into program language code.
- L. The student will use arithmetic, relational, and logical operators to control processing.
- M. The student will control the readability of program output through formatting.
- N. The student will become familiar with the array data structure.
- O. The student will use alphanumeric (combined alphabetic and numeric) and numeric data.

**II. Content**

4.2, 4.3, 4.4, 4.5

- A. Microcomputer Environment
  - 1. Physical set up – keyboard and printer
  - 2. Architecture --  
CPU, ROM, RAM, and interface
  - 3. Proper use of equipment
  
- B. Basic Elements of a Program
  - 1. Input
  - 2. Process
  - 3. Output
  
- C. Program Readability
  - 1. Style
  - 2. Modularity
  - 3. Documentation
  - 4. Flexibility
  - 5. Efficiency
  
- D. Variables and Constants
  - 1. Built-in Data Types (int, double, long, char, bool)
  - 2. String
  - 3. Expressions and Operators (PEMDAS)
  
- E. Controlling Program Flow
  - 1. Evaluation of boolean expressions
  - 2. Sequential (if, else, else-if statements)
  - 3. Logical Operators
  - 4. Looping (do-while, while, for loops)
  
- F. Program Design
  - 1. Structure Charts/Flow Charts
  - 2. Function Decomposition
  - 3. Top-down
  - 4. Algorithms
  
- G. Debugging
  - 1. Tracing
  - 2. System Debuggers (optional)
  
- H. Functions
  - 1. Parameter passing
  - 2. Overloaded
  - 3. Building a Library
  - 4. Mathematical
  - 5. Recursive
  
- I. Classes and Objects
  - 1. Constructors
  - 2. Functions
  - 3. Using Classes
  - 4. String Class

- J. Data Structures
  - 1. Array
  - 2. Matrix (optional)
  
- K. Standard Algorithms
  - 1. Searching
  - 2. Sorting
  
- L. Applications of Programming Techniques
  - 1. Algebraic
  - 2. Geometric
  - 3. Business

III. Materials

- A. Text: Guide to Programming in C++, Corica : Lawrenceville Press

IV. Evaluation

- A. The student will be expected to complete classwork, homework, keep a notebook and take tests and quizzes. These will be checked and reviewed by the teacher.
- B. The student will be expected to demonstrate an acceptable level of proficiency in the objectives and content of this course.
- C. The student will be expected to demonstrate at all times appropriate classroom behavior such as self-control, respect for others, respect for property and a mature attitude.
- D. The student will be expected to adhere to the school rules and regulations for behavior and the district policy for attendance.
- E. Students will be required to successfully pass the High School Proficiency Assessment as mandated in the graduation law (N.J.S.A. 6:8-4.2).
- F. Students who fail the HSPA examination will be placed in a Basic Skills Math class as required by N.J.S.A. 6:8-4.2. There will be no exceptions to this requirement.
- G. The student will be expected to take a comprehensive final exam covering the entire school year's work. This exam will count at 1/5 of the final grade.
- H. The final grade represents the teacher's professional judgment of the student's performance and all of the aforementioned activities and/or requirements are included in the evaluative process.