COMPUTER PROGRAMMING 1 with C++
COURSE CODE: 5056

COURSE DESCRIPTION: This course of study is designed to emphasize the fundamentals of computer programming. Topics include computer software, program design and development, and practical experience in programming, using modern, object-oriented languages.

OBJECTIVE: Given the necessary equipment, software, supplies, and facilities, the student will be able to successfully complete the following core standards for courses that grant one unit of credit.

PREREQUISITE: Any computer related course, Algebra I (or equivalent), and/or teacher recommendation

COMPUTERS REQUIRED: One computer per student

CREDIT: 1 unit per course

RECOMMENDED GRADE LEVEL: 9-12

Course Suggestions:

Students planning to major in Computer Science or Computer Engineering would more likely use Java. According to the State Standards Committee, movement is away from C++ and toward Java. XML, C#, COM, and other newer technologies such as .NET could also be addressed as time, interest, and teacher training permit. This may vary across the state, and local advisory committees should be involved in deciding which language(s) to offer in a particular district/school.

A. SAFETY AND ETHICS

1. Identify major causes of work-related accidents in offices.
2. Describe the threats to a computer network, methods of avoiding attacks, and options in dealing with virus attacks.
3. Identify potential abuse and unethical uses of computers and networks.
4. Explain the consequences of illegal, social, and unethical uses of information technologies, e.g., piracy; illegal downloading; licensing infringement; and inappropriate uses of software, hardware, and mobile devices.
5. Differentiate between freeware, shareware, and public domain software copyrights.
6. Discuss computer crimes, terms of use, and legal issues such as copyright laws, fair use laws, and ethics pertaining to scanned and downloaded clip art images, photographs, documents, video,

recorded sounds and music, trademarks, and other elements for use in Web publications.
7. Identify netiquette including the use of email, social networking, blogs, texting, and chatting.
8. Describe ethical and legal practices in business professions such as safeguarding the confidentiality of business-related information.

B. EMPLOYABILITY SKILLS

1. Identify positive work practices, e.g., appropriate dress code for the workplace, personal grooming, punctuality, time management, and organization.
2. Demonstrate positive interpersonal skills, e.g., communication, respect, and teamwork.

C. STUDENT ORGANIZATIONS

1. Explain how related student organizations are integral parts of career and technology education courses.
2. Explain the goals and objectives of related student organizations.
3. List opportunities available to students through participation in related student organization conferences/competitions, community service, philanthropy, and other activities.
4. Explain how participation in career and technology education student organizations can promote lifelong responsibility for community service and professional development.

D. COMPUTER SYSTEMS

1. Define what a computer is and its purpose.
2. Define basic computer terminology.
3. Define basic programming terminology.
4. Identify basic hardware and software components.
5. Explain the flow of data and instructions through the computer system.
6. Identify components of the programming development environment.
7. Describe the concept of OOP (object-oriented programming).

E. PROGRAM DOCUMENTATION

1. Describe the purpose and value of the program.
2. Define the input for the program.
3. Define the output of the program.
4. Define variables and constants associated with the program using descriptive names and appropriate data types associated with the program.

5. Describe the scope of variables.

F. PROGRAMMING DESIGN

1. List in sequence the steps for developing a program.
2. Develop an algorithm (pseudocode) for a program.
3. Key the program.
4. Save the program.
5. Execute the program.
6. Debug the program for errors (e.g., syntax, run-time, and logic).
7. Run the program to test the logical validity of an application program given appropriate data.

G. PROGRAMMING

1. Describe the purpose/function of different objects.
2. Describe the purpose/function of an event procedure.
3. Identify correctly written Property assignment statements.
4. Demonstrate proper code commenting/documentation techniques.
5. List and define arithmetic, relational, and logical/boolean operators.
6. Explain operator precedence.
7. Differentiate between commands and statements.
8. Write valid variable and constant declaration statements using appropriate data types.
9. Write valid variable and constant declaration statements using appropriate scope (e.g., local, global, static).
10. Write a program that will perform calculations on given data.
11. Write an interactive program that includes features to get input and provide feedback/information (e.g, alerts, messages, input boxes).
12. Identify different decision structures that control program flow.
13. Use built-in functions to generate random numbers.
14. Write a program using accumulators and counters.
15. Identify different looping/iteration structures that control program flow.
16. Use built-in properties and functions to manipulate classes and structures (e.g., String, Math).
17. Describe the conversion from ASCII/Unicode to Hexadecimal and Binary.
18. Describe the purpose/function of general sub procedures.
19. Describe the purpose/function of arguments and parameters.
20. Describe the purpose/function of function procedures.
21. Write a program using one or more general sub procedures and/or functions.
22. Write a program that passes arguments to another general sub procedure and/or function.