

COMPUTER PROGRAMMING 2 WITH C++
COURSE CODE: 5057
STUDENT PROFILE

STUDENT'S NAME:		TEACHER'S NAME:			
School Year/Semester:		Grade:			
Begin Date:		Date Completed:			
<p>Directions: Document student's progress using the applicable rating scales below: Enter date of completion under the appropriate column.</p> <p>0 - Has not received instruction in this area / no experience or knowledge of this task (N/A)</p> <p>1 – Can apply and perform independently (80-100)</p> <p>2 – Can perform the task completely with limited supervision (70-79)</p> <p>3 – Requires additional instruction and or close supervision (60-69)</p>					
A. SAFETY		0	1	2	3
1	Review school safety policies and procedures.				
2	Review classroom safety rules and procedures.				
3	Review safety procedures for using equipment in the classroom.				
4	Identify major causes of work-related accidents in office environments.				
5	Demonstrate safety skills in an office/work environment.				
B. STUDENT ORGANIZATIONS		0	1	2	3
1	Identify the purpose and goals of a Career and Technology Student Organization (CTSO).				
2	Explain how CTSOs are integral parts of specific clusters, majors, and/or courses.				
3	Explain the benefits and responsibilities of being a member of a CTSO.				
4	List leadership opportunities that are available to students through participation in CTSO conferences, competitions, community service, philanthropy, and other activities.				
5	Explain how participation in CTSOs can promote lifelong benefits in other professional and civic organizations.				
C. TECHNOLOGY KNOWLEDGE		0	1	2	3
1	Demonstrate proficiency and skills associated with the use of technologies that are common to a specific occupation				
2	Identify proper netiquette when using e-mail, social media, and other technologies for communication purposes.				

3	Identify potential abuse and unethical uses of laptops, tablets, computers, and/or networks.				
4	Explain the consequences of social, illegal, and unethical uses of technology (e.g., cyber bullying; piracy; illegal downloading; cyberbullying; licensing infringement; inappropriate uses of software, hardware, and mobile devices in the work environment).				
5	Discuss legal issues and the terms of use related to copyright laws, fair use laws, and ethics pertaining to downloading of images, photographs, Creative Commons, documents, video, sounds, music, trademarks, and other elements for personal use.				
6	Describe ethical and legal practices of safeguarding the confidentiality of business-related information.				
7	Describe possible threats to a laptop, tablet, computer, and/or network and methods of avoiding attacks.				
D. PERSONAL QUALITIES AND EMPLOYABILITY SKILLS		0	1	2	3
1	Demonstrate punctuality.				
2	Demonstrate critical thinking and problem-solving skills				
3	Demonstrate initiative and self-direction.				
4	Demonstrate integrity.				
5	Demonstrate work ethic.				
6	Demonstrate conflict resolution skills.				
7	Demonstrate listening and speaking skills.				
8	Demonstrate respect for diversity.				
9	Demonstrate customer service orientation.				
10	Demonstrate teamwork.				
E. PROFESSIONAL KNOWLEDGE		0	1	2	3
1	Demonstrate global or “big picture” thinking.				
2	Demonstrate career and life management skills and goal-making.				
3	Demonstrate continuous learning and adaptability skills to changing job requirements.				
4	Demonstrate time and resource management skills.				
5	Demonstrates information literacy skills.				
6	Demonstrates information security skills.				

7	Demonstrates information technology skills.				
8	Demonstrates knowledge and use of job-specific tools and technologies.				
9	Demonstrate job-specific mathematics skills.				
10	Demonstrates professionalism in the workplace.				
11	Demonstrates reading and writing skills.				
12	Demonstrates workplace safety.				
F. SECURITY AND ETHICS		0	1	2	3
1	Evaluate how sensitive data can be affected by malware and other attacks (e.g., denial-of-service attacks, ransomware, viruses, worms, spyware, phishing) (CSTA, 2017).				
2	Identify best practices of software development that improve computer security and protect devices and information from unauthorized access (e.g., encryption, authentication strategies, secure coding, safeguarding keys). (CSTA, 2017)				
3	Explain how to document programs that use non-user-created resources (e.g., code, media, libraries) giving attribution to the original creator. (Ethical and fair use)				
4	Examine licenses (i.e., permissions) that limit or restrict use of resources (e.g., freeware, shareware, open source, creative commons)				
5	Discuss the implications of using licensed resources in a developed solution.				
G. INTERMEDIATE PROGRAM DESIGN & DEVELOPMENT		0	1	2	3
<u>Problem-Solving and Program Design</u>					
1	Investigate how a problem is systematically solved through the selection and integration of hardware and software components.				
2	Justify hardware and software selections for specific applications by evaluating the components (e.g., databases, sensors, application programming interfaces (APIs)) of various computing devices (e.g., desktops, laptops, tablets, smartphones, and specialized devices like global positioning systems (GPSs)).				
3	Compare and contrast ways software developers protect both devices and information from unauthorized access (e.g., encryption, authentication strategies, secure coding, safeguarding keys) (CSTA, 2017).				
4	Evaluate existing computing solutions according to inclusivity or non-inclusivity (e.g., sight and hearing impairment, ethnicity, age).				
5	Evaluate algorithms in terms of efficiency, correctness, and clarity (CSTA, 2017).				

6	Categorize a variety of algorithms (e.g. linear, exponential, logarithmic, regression etc.).				
<u>Data Structures</u>					
1	Compare and contrast the various data collection methods, data analysis tools, and data representation tools.				
2	Compare and contrast the various data storage tools and data organization methods.				
3	Compare and contrast the various data collection methods, data analysis tools, and data representation tools.				
4	Determine when data structures (e.g., lists, arrays, tuples, stacks, queues, structures) are more appropriate than simple data types, and incorporate them into programs.				
5	Create and modify data structures (e.g., lists, arrays, tuples, stacks, queues, structures) are more appropriate than simple data types, and incorporate them into programs.				
<u>Control Structures</u>					
1	Justify the selection of specific programming constructs, explaining the benefits and drawbacks of choices made on the program's execution. (e.g., procedures, modules, objects).				
2	Build a complex solution to a problem that incorporates reusable code (e.g., student created, application programming interfaces (APIs), libraries).				
3	Justify the selection of specific control structures explaining the benefits and drawbacks of choices made (e.g., trade-offs involving implementation, readability, and program performance). (e.g., conditionals, loops, event handlers, recursion).				
4	Implement and modify built-in classes.				
5	Create and use user-defined classes.				
6	Instantiate and use objects from user-defined classes.				
<u>Testing, Debugging, and Revisions</u>					
1	Recommend modifications for existing computing devices and software to improve functionality for end users.				
2	Evaluate and refine programs to make them more usable, functional, and accessible.				
<u>Documentation</u>					
1	Implement version control to track program refinements.				
2	Document programs that use non-user-created resources (e.g., code, media, libraries) giving attribution to the original creator.				