SEACOAST SCHOOL OF TECHNOLOGY

Computer Science Competencies

Technical Competencies
1. Perform analysis of application requirements to develop a computer program.
2. Perform program design functions in developing an application that meets specified requirements.
3. Develop algorithms to implement program design.
4. Write a technical description of the tasks that the software program performs individually and as part of a team.
5. Discuss and demonstrate knowledge of the Software Development Life Cycle.
6. Use software debugging tools and techniques to insure verification of programs operations both individually and as part of a software development team.
7. Demonstrate the ability to design an intuitive software user interface that possesses a high degree of usability.
8. Discuss and demonstrate knowledge of the program creation process including the concepts of source code, object code, and executable code.
9. Demonstrate the ability to write computer programs using both compiled and interpreted programming languages.
10. Discuss and apply fundamental concepts of programming language..
11. Demonstrate and use a variety of software development tools for program implementation.
12. Write computer programs utilizing the structured programming paradigm.
13. Write computer programs utilizing the object oriented programming paradigm.
14. Write programs using modularization techniques to reduce program complexity and improve program maintainability.
15. Discuss and demonstrate the fundamental level of object oriented design principles including the use of classes and objects in the context of program design.
16. Discuss and demonstrate the fundamental level of instantiation, encapsulation, inheritance, and polymorphism as it applies to object oriented program design.
17. Write a program that involves the design and implementation of a custom class.
18. Discuss and demonstrate knowledge of the relationship between class definition and a class implementation.
19. Write programs that use events to cause program execution to react to the event by writing the appropriate event handler code.
20. Write programs that use a graphical user interface to provide user interaction with a program.
21. Discuss and develop a good user interface design.
22. Conduct usability testing of software.
23. Write programs that access external data files.
24. Write programs that input from and output to external devices.
25. Discuss and demonstrate the different file formats and structures.
26. Write programs that sort data.
27. Write programs that search data.
28. Write programs to solve mathematical problems through numerical analysis concepts.
29. Demonstrate and write programs that simulate physical processes.
30. Demonstrate general problem solving techniques to solve a variety of computational problems.
31. Discuss computer security and its relationship to the computer programmer.
32. Explain and demonstrate principles of computer networks.
33. Discuss the ethical issues involved in computer programming.
34. Discuss computer hacking and cracking and how it relates to the computer programmer.
Core Competencies
35. Discuss and assess venture creation possibilities and identify the steps in planning the venture.
36. Identify the resources needed for venture startup and operation.
37. Discuss the options in planning the venture's future (growth, development, demise).
38. Identify and discuss the traits and behaviors of an entrepreneur (leadership, personal assessment, personal management).
39. Demonstrate personal growth, community leadership, democratic principles and social responsibility by participating in activities/events offered through student organizations.
40. Decision-Making & Problem-Solving
41. Self-Management
42. Communication Skills
43. Ability to Work with Others
44. Information Use – Research, Analysis & Technology
45. Mathematical Concepts
46. General Safety
47. Career Development