

Theory and Foundation

Overall Expectations

- TFV.01 · describe the steps in the software life cycle (problem definition, analysis, design, implementation, testing, and maintenance);
- TFV.02 · explain data structures and their processing algorithms;
- TFV.03 · analyse a number of programming paradigms;
- TFV.04 · explain the importance of program correctness and efficiency;
- TFV.05 · describe the relationship among hardware, software, and network requirements.

Problem Solving, Logic, and Design

- TF1.01 – describe the components of the software life cycle and their importance in project settings;
- TF1.02 – explain the importance of designing reusable code for large software projects;
- TF1.03 – identify similarities and differences among data structures, including arrays, records, and arrays of records, and their applicability to solving programming problems;
- TF1.04 – evaluate the efficiency of different algorithms and their applicability to solving the same programming problem;
- TF1.05 – describe the difference between procedural and object-oriented programming;
- TF1.06 – explain the levels of program correctness: syntax errors, runtime errors, valid data, invalid data, robustness.

Programming Concepts

- TF2.01 – describe how procedural and object-oriented programming paradigms can be used to solve different problems;
- TF2.02 – describe how user-defined types and records provide more flexible and powerful ways of handling data;
- TF2.03 – explain how recursion can be used to solve specific kinds of computing problems.

Hardware, Interfaces, and Networking Systems

- TF3.01 – explain the role of a network in accessing computer software resources;
- TF3.02 – describe the issues involved in maintaining a software library (e.g., access, backup, version control);
- TF3.03 – relate hardware requirements to user software demands.

Skills and Processes

Overall Expectations

- SPV.01 · incorporate the software life cycle in project settings;
- SPV.02 · effectively use software development and diagnostic tools;
- SPV.03 · implement advanced data structures and algorithms;
- SPV.04 · identify on-line and off-line resource materials;
- SPV.05 · use file management techniques in project settings.

Problem Solving, Logic, and Design

- SP1.01 – devise a plan for a large software project (e.g., an accounts receivable or a random walker program), outlining the required activities at each stage of the software life cycle;
- SP1.02 – use industry-standard methodology (e.g., flow chart, pseudocode, structure chart) in the design process;

- SP1.03** – incorporate modularity, software reuse, and maintenance considerations at the design and implementation stages of the project;
- SP1.04** – incorporate appropriate code from shared software libraries into software projects;
- SP1.05** – select appropriate data structures (e.g., arrays, records, arrays of records) for use in projects;
- SP1.06** – design algorithms to incorporate data structures in projects;
- SP1.07** – ensure program correctness by developing a complete suite of test data (valid and invalid data) to eliminate syntax, runtime, and logic errors;
- SP1.08** – use a problem-solving protocol to troubleshoot computer programs.

Programming Practices

- SP2.01** – use an integrated development environment to create and manage a project;
- SP2.02** – employ user-defined data types and record data types to improve program efficiency;
- SP2.03** – use arrays, records, and arrays of records in different project settings;
- SP2.04** – build and maintain a small software library to facilitate the reuse of code;
- SP2.05** – incorporate appropriate maintenance considerations during the implementation of programs;
- SP2.06** – use recursion in a simple program;
- SP2.07** – compare the effectiveness of several algorithms for solving the same problem;
- SP2.08** – produce comprehensive documentation (e.g., help files, manuals) for a software project;
- SP2.09** – perform peer reviews of internal and external documentation;
- SP2.10** – perform line-by-line walk-throughs of computer programs that include all program structures;
- SP2.11** – use appropriate research and resource materials to independently master new programming skills;
- SP2.12** – effectively critique programs written by others;
- SP2.13** – log error messages and appropriate fixes.

Hardware, Interfaces, and Networking Systems

- SP3.01** – implement a backup strategy for program files on different media;
- SP3.02** – develop software libraries in project settings;
- SP3.03** – use predefined modules from software libraries to improve productivity.

Impact and Consequences

Overall Expectations

- ICV.01** · describe issues related to the ethical use of computers;
- ICV.02** · describe the use of information technology and its impact in the community;
- ICV.03** · identify postsecondary educational opportunities leading to careers in information systems and computer science;
- ICV.04** · explain the importance of employability skills and lifelong learning to information technology careers.

The Ethical Use of Computers

- IC1.01** – explain the importance of the ethical use of computers in areas such as software piracy, privacy, and security;
- IC1.02** – describe the essential elements of a code of computing ethics and why it is important to have and follow such a code;

IC1.03 – analyse current media information relating to ethical issues in computing.

Effects of Information Technology

IC2.01 – describe how local industries, businesses, or community groups are affected by the growing use of information technology to facilitate communication;

IC2.02 – describe, using presentation software, how local industries, businesses, or community groups use computers to improve efficiency and productivity to serve their clients;

IC2.03 – evaluate the pros and cons of moving to new hardware and software technologies (e.g., costs, training requirements, compatibility, deployment);

IC2.04 – use appropriate strategies to avoid potential health and safety problems associated with computer use, such as musculo-skeletal disorders and eye strain.

Postsecondary Education, Career Opportunities, and Employability Skills

IC3.01 – describe the range of career opportunities in computing and their lifelong learning requirements;

IC3.02 – produce job descriptions for occupations/ professions in computer and information science;

IC3.03 – demonstrate communication skills (e.g., the ability to provide comprehensive internal documentation and the ability to explain program design and implementation clearly) in a team setting;

IC3.04 – describe the elements of working effectively in a team environment (e.g., conflict resolution, time management, constructive criticism, task assignment).