

Theory and Foundation

Overall Expectations

- TFV.01** · describe at least two problem-solving models;
- TFV.02** · identify the stages in the software development process (problem definition, analysis, design, implementation, testing, and maintenance);
- TFV.03** · explain standard control and data structures used in computer programs;
- TFV.04** · identify on-line and off-line resources;
- TFV.05** · explain the functions of basic computer components.

Problem Solving, Logic, and Design

- TF1.01** – define problems by identifying the expected output and necessary user input;
- TF1.02** – evaluate the usefulness of available software tools in a problem-solving situation, using criteria such as ease of use and time required for processing;
- TF1.03** – describe the steps in the software development process and their importance in the development of large programs or applications;
- TF1.04** – explain different problem-solving models (e.g., top-down, bottom-up) that can be used to create a computer program;
- TF1.05** – determine the level of error checking required for given problems;
- TF1.06** – identify the possibilities and limitations of proposed designs;
- TF1.07** – document for the user the potential and limitations of programs.

Programming Concepts

- TF2.01** – describe the characteristics of integer, real, character, and Boolean data types;
- TF2.02** – define constants, variables, expressions, and assignment statements;
- TF2.03** – describe the order in which arithmetic and logical operations are performed;
- TF2.04** – explain the use of Boolean operators in compound expressions;
- TF2.05** – define the structure of one- and two-dimensional arrays and associated concepts (e.g., subscripts, elements, bounds);
- TF2.06** – explain the purpose of selection and repetition structures, and how they are expressed in a programming language;
- TF2.07** – describe the purpose of functions and procedures, and how they are expressed in a programming language;
- TF2.08** – describe parameter passing and scope;
- TF2.09** – identify differences between local and global variables;
- TF2.10** – identify differences among logic, runtime, and syntax errors;
- TF2.11** – describe the evolution of programming languages (e.g., machine, assembly, high-level, 4GL);
- TF2.12** – evaluate available on-line resources such as "readme" files, help files, and "frequently asked questions" files;
- TF2.13** – evaluate available off-line resources such as user manuals and reference manuals;
- TF2.14** – explain the importance of external and internal documentation and programming style;
- TF2.15** – identify common acronyms used in the computing industry.

Hardware, Interfaces, and Networking Systems

- TF3.01** – describe the function and location of the basic components of a computer (e.g., motherboard, CPU, I/O devices, memory);

- TF3.02** – identify common computer peripheral devices (e.g., mouse, keyboard, screen, printer, multimedia devices) and their primary functions;
- TF3.03** – explain differences among software for systems, applications, and programming.

Skills and Processes

Overall Expectations

- SPV.01** · develop effective programs by following the steps in the software design process;
- SPV.02** · use defined programming practices (e.g., headers, indentation, internal documentation, informative variable names);
- SPV.03** · produce appropriate internal and external documentation;
- SPV.04** · properly use an operating system, including a network.

Problem Solving, Logic, and Design

- SP1.01** – resolve ambiguities and missing information in problem statements;
- SP1.02** – use the input, process, and output model to solve problems;
- SP1.03** – select suitable data structures to represent information;
- SP1.04** – develop and maintain a project plan that covers all aspects of the development process for a computer program;
- SP1.05** – develop appropriate algorithms in text or diagram form to solve problems and verify solutions;
- SP1.06** – produce user-friendly input and output forms;
- SP1.07** – solve the same problem using various tools (e.g., a calculator and a computer program, a sort program and a spreadsheet/database/word processor sort function);
- SP1.08** – verify solutions to problems;
- SP1.09** – incorporate modularity into the design process.

Programming Practices

- SP2.01** – use constants, variables, expressions, and assignment statements to store and manipulate numeric, character, and logical data in programs;
- SP2.02** – incorporate one-dimensional and two-dimensional arrays into computer programs;
- SP2.03** – write programs that use related arrays to store and extract data;
- SP2.04** – use selection structures, counted and conditional loops, and nested selection and loop structures;
- SP2.05** – manipulate numbers and text using built-in subroutines;
- SP2.06** – write subroutines that pass parameters and use local and global variables;
- SP2.07** – implement a program design using sequence, selection, and repetition structures;
- SP2.08** – use on-line and off-line reference materials effectively;
- SP2.09** – adhere to defined programming style, including naming conventions for variables and subroutines, indentation, and spacing;
- SP2.10** – incorporate and maintain internal documentation to a specific set of standards, including author, date, file name, purpose, and explanatory comments of major statement groups;
- SP2.11** – develop external documentation (including pseudocode, diagrams, and charts) to summarize the design;
- SP2.12** – test completed programs with a full range of valid data to ensure that all components work as expected;
- SP2.13** – interpret errors during testing and program execution;

- SP2.14** – trace program execution using manual methods and software debugging tools;
- SP2.15** – identify and correct logic, runtime, and syntax errors in programs;
- SP2.16** – use linear searches and simple sort routines in programs;
- SP2.17** – write programs that access sequential files;
- SP2.18** – perform peer evaluations of internal documentation and programming style.

Hardware, Interfaces, and Networking Systems

- SP3.01** – use an operating system to perform tasks such as managing files and configuring hardware;
- SP3.02** – use built-in networking functions such as shared files and input/output devices;
- SP3.03** – use common Internetworking services to access and navigate global information resources;
- SP3.04** – develop computer resources to share information globally or locally;
- SP3.05** – implement a comprehensive backup strategy for files.

Impact and Consequences

Overall Expectations

- ICV.01** · explain issues related to the ethical use of computers;
- ICV.02** · describe emergent technologies and their impact on society;
- ICV.03** · identify information systems and computer science career paths, and their educational requirements.

Effects of Information Technology

- IC1.01** – explain how the pervasiveness of computer technology affects daily life;
- IC1.02** – describe how information is gathered using computers and how this can affect peoples' privacy and right to information;
- IC1.03** – identify a number of available sources of career and educational information using networks and evaluate their reliability and accuracy;
- IC1.04** – describe, using presentation software, emergent technologies and their potential influence on society;
- IC1.05** – use appropriate strategies to avoid potential health and safety problems associated with computer use, such as musculo-skeletal disorders and eye strain;
- IC1.06** – explain the importance to identifying career paths of keeping up to date on current articles and thought on computer technology.

Postsecondary Education and Career Opportunities

- IC2.01** – identify postsecondary educational opportunities leading to careers in information systems and computer science, and report on their entry requirements;
- IC2.02** – identify which careers require computer expertise, using local or national media;
- IC2.03** – identify opportunities for apprenticeship training and co-op programs.