

**CONTENT STANDARD 1.0: UNDERSTAND PROGRAMMING PRINCIPLES**

**Performance Standard 1.1: Demonstrate Critical Thinking and Problem-Solving Skills as they Apply to Programming**

- 1.1.1 Apply basic programming principles.
- 1.1.2 Describe and differentiate procedural and object-oriented programming.
- 1.1.3 Apply the features of object-oriented programming languages.
- 1.1.4 Write a program that produces output.
- 1.1.5 Select identifiers to use within programs.
- 1.1.6 Improve programs by adding comments.
- 1.1.7 Write and run a program.

**CONTENT STANDARD 2.0: PROBLEM SOLVING THROUGH PROGRAMMING**

**Performance Standard 2.1: Demonstrate Ability to Use Variables, Data Types, and String Manipulation to Solve Computer Problems Programmatically**

- 2.1.1 Demonstrate the process of declaring variables.
- 2.1.2 Display variable values.
- 2.1.3 Apply integral data types.
- 2.1.4 Apply floating-point data types.
- 2.1.5 Apply arithmetic operators.
- 2.1.6 Apply Boolean data type.
- 2.1.7 Apply numeric type conversion.
- 2.1.8 Apply char data type.
- 2.1.9 Apply string data type.
- 2.1.10 Define named constants and enumerations.

**CONTENT STANDARD 3.0: USE LOGIC IN PROGRAMMING**

**Performance Standard 3.1: Demonstrate Effective Use of Selection Structures to Add Logic to Programs**

- 3.1.1 Demonstrate logic-planning tools and decision-making.
- 3.1.2 Make decision using the “if” statement.
- 3.1.3 Make decisions using the if-else statement.
- 3.1.4 Apply compound expressions in if statements.
- 3.1.5 Make decisions using the switch statement.
- 3.1.6 Apply the conditional operator.
- 3.1.7 Apply the NOT operator.
- 3.1.8. Describe how to avoid common errors when making decisions, and apply problem-solving skills in context.

#### **CONTENT STANDARD 4: PROGRAMMING AND VALIDATION**

##### **Performance Standard 4.1: Demonstrate Ability to Test, Debug and Validate Programming Applications**

- 4.1.1 Locate a logic error by stepping through the code.
- 4.1.2 Locate logic errors using breakpoints.
- 4.1.3 Fix syntax and logic errors.
- 4.1.4 Select appropriate test data for an application.

#### **CONTENT STANDARD 5.0: UNDERSTAND REPETITION IN PROGRAMMING**

##### **Performance Standard 5.1: Differentiate Between the Various Types of Repetition**

- 5.1.1 Apply the loop structure.
- 5.1.2 Create loops using the while statement.
- 5.1.3 Create loops using the for statement.
- 5.1.4 Create loops using the do statement.
- 5.1.5 Apply nested loops.
- 5.1.6 Apply accumulators.
- 5.1.7 Understand and describe how to improve loop performance

#### **CONTENT STANDARD 6.0: DEMONSTRATE PROGRAMMING FUNCTIONALITY**

##### **Performance Standard 6.1: Use Methods to Increase Functionality and to Modularize Programs**

- 6.1.1 Describe methods and implementation hiding.
- 6.1.2 Write methods with no parameters and no return value.
- 6.1.3 Write methods that require a single argument.
- 6.1.4 Write methods that require multiple arguments.
- 6.1.5 Write a method that returns a value.
- 6.1.6 Pass an array to a method.
- 6.1.7 Overload methods.
- 6.1.8 Demonstrate how to avoid methods.
- 6.1.9 Apply optional parameters.

#### **CONTENT STANDARD 7.0: UNDERSTAND ARRAYS AND STRUCTURE CONCEPTS**

##### **Performance Standard 7.1: Demonstrate Understanding of Arrays and Structure and Apply Concepts In Program Development**

- 7.1.1 Declare an array and assign values to array elements.
- 7.1.2 Access array elements.
- 7.1.3 Search an array using a loop.
- 7.1.4 Apply multidimensional arrays.

**CONTENT STANDARD 8.0: UNDERSTAND CLASSES IN PROGRAMMING**

**Performance Standard 8.1: Students will demonstrate understanding of Object-Oriented Programming Concepts**

- 8.1.1 Describe and apply class concepts.
- 8.1.2 Create classes from which objects can be instantiated.
- 8.1.3 Create objects.
- 8.1.4 Create properties, including auto-implemented properties.
- 8.1.5 Use public fields and private methods.
- 8.1.6 Define the "this" reference.
- 8.1.7 Write constructors.
- 8.1.8 Use object initializers.
- 8.1.9 Overload operators.
- 8.1.10 Declare an array of objects.
- 8.1.11 Use sorting methods with an array of objects.
- 8.1.12 Write destructors.
- 8.1.13 Describe and demonstrate inheritance.
- 8.1.14 Extend classes.
- 8.1.15 Override base class methods.
- 8.1.16 Describe how a derived class object "is an" instance of the base class.
- 8.1.17 Define the object class.
- 8.1.18 Use base class constructors.
- 8.1.19 Create abstract classes.
- 8.1.20 Create use interfaces.
- 8.1.21 Apply extension methods.
- 8.1.22 Describe the benefits of inheritance.
- 8.1.23 Recognize inheritance in GUI applications.

**CONTENT STANDARD 9.0: UNDERSTAND PROGRAMMING AND EXCEPTIONS**

**Performance Standard 9.1: Demonstrate Exception-Handling in Program Development**

- 9.1.1 Compare and demonstrate traditional and object-oriented error-handling methods.
- 9.1.2 Cast data types.
- 9.1.3 Catch multiple exceptions.
- 9.1.4 Apply the finally block.
- 9.1.5 Handle exceptions thrown from outside methods.
- 9.1.6 Trace exceptions through the call stack.
- 9.1.7 Create exception classes.
- 9.1.8 Re-throw exceptions.

**CONTENT STANDARD 10.0: UNDERSTAND PROGRAMMING AND EVENTS**

**Performance Standard 10.1: Use Event Handlers in Programs**

- 10.1.1 Define and apply event handling.
- 10.1.2 Define and describe delegates.
- 10.1.3 Declare own events and handlers.
- 10.1.4 Use built-in event handlers.
- 10.1.5 Handle control component events.
- 10.1.6 Handle mouse and keyboard events.
- 10.1.7 Manage multiple controls
- 10.1.8 Explain how to find more information on controls and events

**CONTENT STANDARD 11.0: SYSTEMS PLANNING AND DEVELOPMENT**

**Performance Standards 11.1: Apply Concepts and Principles of Systems Planning and Development**

- 11.1.1 Describe the information systems development life cycle (SDLC).
- 11.1.2 Discuss how to evaluate off-the-shelf software.
- 11.1.3 Explain reuse and its role in software development.
- 11.1.4 Describe the skills required to be an effective project manager.  
List and describe the skill and activities of a project manager during project initiation,
- 11.1.5 planning, execution, and closedown.
- 11.1.6 Describe the steps for identifying and selecting projects and initiating and planning projects.
- 11.1.7 Explain the need for and contents of a project scope statement.
- 11.1.8 Compare various methods for assessing project feasibility.

**CONTENT STANDARD 12.0: SYSTEMS ANALYSIS**

**Performance Standards 12.1: Demonstrate Competency with Systems Analysis Tools and Concepts**

- 12.1.1 Compare options for designing and conducting interviews to determine system requirements.
- 12.1.2 Develop a plan for conducting an interview to determine system requirements.
- 12.1.3 Explain the advantages and pitfalls of observing workers and analyzing business documents to determine system requirements.
- 12.1.4 Plan a joint application design session.
- 12.1.5 Use prototyping during requirements determination.
- 12.1.6 Select appropriate methods to elicit system requirements.
- 12.1.7 Describe how requirements determination techniques apply to development of Internet applications.
- 12.1.8 Demonstrate the logical modeling of processes through studying examples of data-flow diagrams, pseudo code, and flowcharts.

**CONTENT STANDARD 13.0: PRINCIPLES OF DESIGN**

**Performance Standards 13.1: Demonstrate Knowledge Of Application Design Principles**

- 13.1.1 Explain the process of designing interfaces and dialogues and the deliverables for their creation.
- 13.1.2 Apply the general guidelines for interface design, including guidelines for layout design, structuring data-entry fields, providing feedback, and system help.
- 13.1.3 Concisely define each of the following key database design terms: relation, primary key, functional dependency, foreign key, referential integrity, field, data type, null value, demoralization, file organization, index, and secondary key.
- 13.1.4 Explain the role of designing databases in the analysis and design of an information system. Transform an entity-relation (E-R) diagram into an equivalent set of well-structured
- 13.1.5 (normalized) relations.
- 13.1.6 Merge normalized relations from separate user views into a consolidated set of well-structured relations.
- 13.1.7 Choose storage formats for fields in database tables.
- 13.1.8 Translate well-structured relations into efficient database tables.
- 13.1.9 Explain when to use different types of file organizations to store computer files.
- 13.1.10 Describe the purpose indexes and the important considerations in selecting attributes to be indexed.

**CONTENT STANDARD 14.0: IMPLEMENTATION AND SUPPORT**

**Performance Standards 14.1: Demonstrate Knowledge of Application Implementation and Identify the Need for Ongoing Application Support**

- 14.1.1 Describe the process of coding, testing, and converting an organizational information system.
- 14.1.2 Outline the deliverables and outcomes of an organizational information system.
- 14.1.3 List the deliverables for documenting the system and for training and supporting users.
- 14.1.4 Compare the many modes available for organizational information system training, including self-training and electronic performance support systems.
- 14.1.5 Discuss the issues of providing support for end users.
- 14.1.6 Explain why application implementation sometimes fails.
- 14.1.7 Describe several factors that influence the cost of maintaining an application.