

# Java Programming

(10155)

## **Rationale Statement:**

The world is full of problems that need to be solved or that need a program to solve them faster. In computer, programming students will learn how to solve story problems and develop a computer program that will solve the problem. Computer programming courses in the state of South Dakota usually are taught using one of three computer languages: Visual Basic, C++, or Java. Students that are interested in the Programming and Software Development pathway will find that taking a computer-programming course in high school will better prepare them for post-secondary work in computer science, engineering, mathematics, and other software development areas.

## **Course Description:**

Grade Level: 10-12

## **Course Topics:**

- Introduction to programming history and the programming language
- Understanding the information processing cycle
- Customer needs analysis for designing a program
- Defining and designing the program project
- Coding an application
- Creating, debugging, and documenting a software application

**NOTE:** The core technical standards and examples are designed for a Programming 1 and Programming 2 Course.

## Core Technical Standards & Examples

<b>Indicator #1: Demonstrate programming as it relates to the customer needs</b>	
<b>Bloom's Taxonomy Level</b>	<b>Standard and Examples</b>
<b>Analyzing</b>	<p><b>JP 1.1 Gather data to identify customer requirements.</b> Examples:</p> <ul style="list-style-type: none"> <li>• Gather information using interviewing strategies.</li> <li>• Identify input, output and system processing requirements.</li> <li>• Clarify specifications using questioning techniques.</li> <li>• Identify hardware, networking, and software system functional requirements.</li> <li>• Demonstrate knowledge of nonfunctional requirements.</li> <li>• Use customer satisfaction in determining product characteristics.</li> </ul>
<b>Applying</b>	<p><b>JP 1.2 Demonstrate knowledge of programming language concepts.</b> Examples:</p> <ul style="list-style-type: none"> <li>• Demonstrate knowledge of the concept of physical representation of digitized information.</li> <li>• Demonstrate knowledge of the hardware-software connection.</li> <li>• Demonstrate knowledge of the function and operation of compilers and interpreters.</li> <li>• Demonstrate knowledge of current key programming languages and the environment they are used in.</li> </ul>
<b>Evaluating</b>	<p><b>JP 1.3 Develop software requirements specification.</b> Examples:</p> <ul style="list-style-type: none"> <li>• Demonstrate knowledge of the use, structure, and contents of a requirements specification document.</li> <li>• Define system and software requirements.</li> <li>• Define business problem to be solved by the application</li> <li>• Develop informal specifications.</li> <li>• Develop formal specification.</li> <li>• Review and verify specification with customer.</li> </ul>

**Indicator #2: Produce IT-based strategies and project plans to solve the problem.**

<b>Bloom's Taxonomy Level</b>	<b>Standard and Examples</b>
<b>Understanding</b>	<b>JP 2.1 Define scope of work for the programming project.</b> Examples: <ul style="list-style-type: none"><li>• Demonstrate knowledge of the key functions and subsystems of the software product.</li><li>• Demonstrate knowledge of software development process and issues.</li><li>• Develop implementation plan.</li></ul>
<b>Applying</b>	<b>JP 2.2 Demonstrate knowledge and skills of working in a software development team.</b> Examples: <ul style="list-style-type: none"><li>• Identify resources and risks.</li><li>• Demonstrate knowledge of cross-functional team structures and team members' roles.</li></ul>

**Indicator #3: Demonstrate knowledge of the software development process.**

<b>Bloom's Taxonomy Level</b>	<b>Standard and Examples</b>
<b>Understanding</b>	<b>JP 3.1 Demonstrate knowledge of software development methodology.</b> Examples: <ul style="list-style-type: none"><li>• Demonstrate knowledge of system analysis issues related to design, testing, implementation, and maintenance.</li><li>• Identify roles on team members/customers in the software development process.</li><li>• Demonstrate knowledge of how to use software methodologies to analyze a real-world problem.</li><li>• Identify constraints of the current project.</li><li>• Demonstrate knowledge of modeling and analyzing functional requirements (e.g., dataflow diagrams, process specifications, and a data dictionary).</li><li>• Demonstrate knowledge of modeling and analyzing data requirements (e.g., Jackson diagrams, entity relationship diagrams, and relations).</li></ul>
<b>Applying</b>	<b>JP 3.2 Apply tools for developing software applications.</b> Examples: <ul style="list-style-type: none"><li>• Demonstrate knowledge of software development environment.</li><li>• Use prototyping techniques.</li><li>• Use desk checking</li><li>• Demonstrate knowledge of reuse and components.</li></ul>
<b>Applying</b>	<b>JP 3.3 Apply language specific programming tools/techniques.</b> Examples: <ul style="list-style-type: none"><li>• Develop programs using appropriate language.</li><li>• Make use of appropriate development environment for the selected language.</li></ul>

**Indicator #4: Create a logical design for a software application.**

<b>Bloom's Taxonomy Level</b>	<b>Standard and Examples</b>
<b>Evaluating</b>	<p><b>JP 4.1 Create design specification for a computer application.</b> Examples:</p> <ul style="list-style-type: none"><li>• Analyze real world problems for the applicability of structured, object oriented, event-driven logical design methods.</li><li>• Design system input, output, processing, and interfaces.</li></ul>
<b>Applying</b>	<p><b>JP 4.2 Analyze real world problems for the applicability of structured, object orientate, even driven logical design methods.</b> Examples:</p> <ul style="list-style-type: none"><li>• Demonstrate knowledge of the characteristics and the uses of processing</li><li>• Identify basic concepts of algorithm and data structure development.</li><li>• Demonstrate knowledge of different data types</li><li>• Identify constraints.</li><li>• Demonstrate knowledge of nonfunctional requirements</li><li>• Demonstrate knowledge of modular design concepts.</li><li>• Demonstrate knowledge of how to design and implement programs in a top-down manner.</li><li>• Analyze and prepare logic using program flowchart.</li><li>• Identify standards and issues related to I/O programming and design of I/O interfaces.</li></ul>

**Indicator #5: Create a computer application by writing code.**

<b>Bloom's Taxonomy Level</b>	<b>Standard and Examples</b>
<b>Applying</b>	<b>JP 5.1 Demonstrate knowledge of programming language concepts.</b> Examples: <ul style="list-style-type: none"><li>• Demonstrate knowledge of the basics of structured, object-oriented, and event-driven programming.</li><li>• Demonstrate knowledge of the concepts of data and procedural representation.</li></ul>
<b>Applying</b>	<b>JP 5.2 Develop an application using selected programming language.</b> Examples: <ul style="list-style-type: none"><li>• Translate logical design into code in an appropriate language argument.</li><li>• Demonstrate knowledge of specific language syntax.</li></ul>
<b>Evaluating</b>	<b>JP 5.3 Demonstrate knowledge of basic software systems implementation.</b> Examples: <ul style="list-style-type: none"><li>• Compile and debug code.</li><li>• Prepare code documentation.</li><li>• Conduct code walkthrough and/or inspection.</li><li>• Troubleshoot unexpected results.</li><li>• Access needed information using company and manufacturers' references.</li></ul>