**J. Sargeant Reynolds Community College**
**Course Content Summary**

**Course Prefix and Number: <u>CSC 221</u>    Credits: <u>3</u>**

**Course Title: <u>Introduction to Problem Solving and Programming</u>**

**Course Description:**

Introduces problem solving and implementation of solutions using a high level programming language in a structured programming environment. Includes concepts and practice of structured programming, problem-solving, top-down design of algorithms, a high level programming language syntax, control structures, arrays, and an introduction into object oriented programming.  First course in a three course sequence. (CSC 221-222-223) The assignments in this course require mathematical problem solving skills, algebraic modeling and functions, and use of variables. Lecture 3 hours. Total 3 hours per week. 3 credits

**General Course Purpose:**

CSC 221, CSC 222, and CSC 223 comprise the standard sequence of minimal programming content for computer science majors. The course sequence will teach the students to use high-level languages and their applications to problem solving by using algorithms within procedural and object-oriented techniques, while ensuring data adheres to a structured model. This course is the first course in the sequence. It introduces computer based problem solving and implementation of solutions in a high level programming language.  Python is the preferred language for this course, institutions may offer using a different language to align with primary 4-year partner requirements.

**Course Prerequisites/Corequisites:**

None.

**Course Objectives:**

Upon completing the course, the student will be able to:

**Civic Engagement**

-

- Perform accurate calculations, interpret quantitative information, apply and analyze relevant numerical data, and use results to support conclusions

**Scientific Literacy**

- Represent real-world objects and processes virtually by identifying properties, behavior, and operations relevant to solving problems on a computer.

**Written Communication**

- Develop, convey, and exchange ideas in writing, as appropriate to a given context and audience

**Basic concepts of computer systems**

- Differentiate computer components by functionality.
- Define basics of computer storage devices.
- Illustrate the computer structure.
- Define Binary and Hexadecimal numeration systems.
- Define types of software.
- Explain the use of computers, and the social impact they have.
- Discuss secure programming
- Evaluate the ethical aspects of programming

**Processing Code**

- Editors, compilers and/or interpreters; distinguishing source code, object code, and executables.
- Reading and evaluate compilation error messages.
- Executing programs.
- Analyzing and resolving run-time errors.

**Problem analysis and algorithmic modeling**

- List and apply the steps involved in problem solving through algorithmic modeling.
- Describe activities related to program modeling and design including algorithm development.
- Solve problems using techniques such as pseudocode, flowcharts, and model development.
- Verify algorithms and identify errors.
- Distinguish between procedural techniques and object-oriented techniques.
- Write programs using good programming practices.

**Use of data**

- Compare and contrast data types.
- Describe the use of variables.
- Build expressions using variables, literal data, and operators, correctly using rules of operator precedence.

**Decision structures**

- Describe how conditional selection operations are used to alter the sequential execution of a program.
- Describe how relational and Boolean operators are used to form logical expressions that evaluate to true or false
- Identify techniques to evaluate selection statements for logic errors.
- Develop programs using sequential and selection operations.

**Repetition structures**

- Describe how repeti (ons)-2 (.)TJ05ns.tionse fofy