# Syllabus

## CSC 115 CS1: Introduction To Programming And Computational Thinking

### General Information

<table>
<thead>
<tr>
<th><strong>Date</strong></th>
<th>April 16th, 2019</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author</strong></td>
<td>William McLaughlin</td>
</tr>
<tr>
<td><strong>Department</strong></td>
<td>Computing Sciences</td>
</tr>
<tr>
<td><strong>Course Prefix</strong></td>
<td>CSC</td>
</tr>
<tr>
<td><strong>Course Number</strong></td>
<td>115</td>
</tr>
<tr>
<td><strong>Course Title</strong></td>
<td>CS1: Introduction To Programming And Computational Thinking</td>
</tr>
</tbody>
</table>

### Course Information

**Catalog Description**  
CS1: Introduction to Programming and Computational Thinking serves as a first course for all computer-related majors. This course is for beginning programmers, and is the first course in a sequence of three programming courses. The course emphasizes the development of languages and software, problem-solving, and programming in a structured, object-oriented language. The Java programming language is used throughout the course.

**Credit Hours**  
3

**Lecture Contact Hours**  
3

**Lab Contact Hours**  
1

**Other Contact Hours**  
0

**Grading Scheme**  
Letter

### Prerequisites

Placement in Math Level 1 (or higher)

### Co-requisites

None

### First Year Experience/Capstone Designation
This course DOES NOT satisfy the outcomes applicable for status as a FYE or Capstone.

SUNY General Education

This course is designated as satisfying a requirement in the following SUNY Gen Ed category
None

FLCC Values

Institutional Learning Outcomes Addressed by the Course
None

Course Learning Outcomes

Course Learning Outcomes

1. Construct fundamental computer algorithms to solve simple problems

2. Create basic computer programs using the formal syntax from a high-level, object-oriented programming language

3. Translate foundational algorithms into simple technical computer program solutions

Outline of Topics Covered

I. Fundamentals of Computer Problem Solving
   • Problem Analysis
   • Design Logic – Simple Algorithmic Development
     I. Flowcharts
     II. Pseudocode

II. Fundamentals of Computer Programming
   • Programming Languages and Environments
     I. Object-Oriented verses Structured Programming and Functional Methodologies
     II. Phases of Language Translation (Compiling, Interpreting, Linking, and Executing)
     III. Java Language Specification: API, JDK, and IDE
     IV. Error Conditions: Syntax, Runtime, and Logic
   • Software Development Process (IPO)
     I.
Requirements

II. Specification

III. Analysis

IV. Design

V. Implementation

VI. Testing

VII. Deployment

VIII. Maintenance

- Creating, Compiling, and Executing a Java Program
  I. Identifiers, Variables, and Constants
  II. Memory Representations and Data Types
     I. Numeric, String, Boolean, Character
  III. Assignment, Numeric, Relational and Logical Operators
  IV. Expression Evaluation: Assignment, Numeric, Boolean
  V. Fundamental Programming Constructs
     I. Sequence
     II. Selection
     III. Iteration

- Subprograms, Functions, and Methods
  I. Formal Parameters, Actual Parameters
  II. Passing Arguments and Return Values
  III. Method Overloading
  IV. Developing Reusable Code

- Secure Coding Techniques
  I. Variable Scope
  II. Input Data Validation

- Arrays
  I. Common Array Operations
  II. Sorting and Searching
This course is required as a core program course in the following program(s)