**Syllabus**
**CS 115**
**Intro Computer Sci I C++**

**Instructor: Jolanta Soltis**
**Email: soltis@njit.edu (Subject : CS--115_ Section)**
**Phone: (973) 596-2925**
**Office Room: WebEx**
**Office Hours: (Webex) Monday and Friday 4:00 pm-5:00 pm**

Textbook:

1. Sign in or create an account at learn.zybooks.com
2. Enter zyBook code
3. **NJITCS115SoltisSpring2024**

**For Windows Visual Studio (download from ist.njit.edu/software/download)**
**For Mac: Visual Studio for Mac or Xcode: http://developer.apple.com/technologies/tools/**
**(Not supported by the instructor)**

**Schedule:** Canvas

**Overview:**
An introduction to the fundamentals of computer science, with emphasis on programming methodology and problem-solving. Topics include basic concepts of computer systems, development environments, software engineering, algorithm design, programming languages, and data abstraction, with applications. A high-level language is fully discussed and serves as the vehicle to illustrate many of the concepts. The programming language used is c++ and the integrated development environment is Visual Studio.

**Topics:**
C++ Basics
Flow of control
Functions
Parameters
Arrays
Structures and Classes
Constructors
Strings
Pointer and dynamic arrays
Inheritance
Polymorphism

**Student Learning Outcomes:**
1. Analyze the given problem statements to create basic program designs.
2. Implement different functions for input and output, various data types, basic operators, files, and functions.
3. Demonstrate basic object-oriented and structured programming concepts.
4. Implement programming techniques to solve problems in the C++ programming language.
5. Apply the concepts and principles of the programming language to real-world problems and solve the problems through project-based learning.

**Grading:**

1. Class Participation and attendance: 5%
2. Labs - 10%
3. Book interactive exercises: 8%
4. Quizzes - 2%
5. Homework challenge activities from the book: 15%
6. Midterm Exams: 25%
7. Final Exam: 35%

**Assignments:**

For all assignments the following timing scheme is valid:

1. In time: 100%
2. One week delay: 90%
3. Two weeks delay: 80%
4. Else: 70%

**Test and Quiz Policy**: No make-up quizzes. The two lowest scores on quizzes will be dropped. Quizzes are usually given at the beginning of a lecture.

**Policy:**

1. All assignments should be submitted by their due date in order to be considered for full credit.
2. can be made for valid medical reasons and family emergencies only. In such a situation documentary evidence is necessary.
3. You could discuss your assignments with others but you have to write all assignments individually.
4. No teamwork allowed.

**Grading:**

**The program will be credited in different levels:**

1. Cannot compile.
2. Can run but the results are not correct.
3. Compiles, runs, produces correct results but has not been adequately tested.
4. Compiles, runs, produces correct results, and has been adequately tested.

**Points will be deducted for the following reason:**

1. Inadequate comments or documentation.
2. Incorrect or bad programming style.
3. Inconsistent indentation.
4. Identifiers names that are not meaningful.

**Homework assignments are due on the specified date. The late assignment will be graded according to the assignment rule listed above.**

**Please submit to Moodle all your homework (if you are sending it to me directly please use sent your homework via email to me with the subject: "CS-115_Homework (1 or 2…) Last Name"**

## Learning Outcomes

Upon completing the course, students will be expected to know and be able to use these elements to compute the solution to a problem:

- Know how to use loops
- Know how to use if statements
- Understand the concept of classes and objects
- Design and implement own classes
- Create and use correctly object of different types
- Devise a sequence of steps (algorithm) that correctly solves a given problem.
- Write a program that implements the algorithm using:
    - The main set of c++ programming language elements (variables, syntax, keywords)
    - Data types (primitive and object data types including arrays)
    - Statements that perform input/output, control statements
    - Exception handling
- Understand inheritance and polymorphism and correctly use to solve complex problems

**Plagiarism:** Plagiarism will not be tolerated and will receive a penalty of an automatic grade of F.

**Resources:**

- http://ist.njit.edu/software/download.php

***Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the academic code of integrity policy that is found at: http://www5.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf.***

## All students must abide by the NJIT honor code.

http://www.njit.edu/academics/honorcode.php