Course Number

CSCI-UA.0101

Course Title

Introduction to Computer Science

Course Description

Formal Course Description

This course will teach students to solve problems by designing algorithms and building them into standalone computer applications. Experience will be acquired through the completion of projects in a high-level programming language. This course is intended for computer science majors but is suitable for students of other scientific disciplines.

Informal Course Description

In this class you will be introduced to the field of computer science. Specifically, you will learn how to design and write computer programs in the Java programming language. You will learn the object oriented paradigm (OOP) and how to model and solve problems using OOP.

You will learn how to apply principles from the fundamentals of computer science using algorithms, data structures, the theory of computing, and the object oriented paradigm to solve real world problems. You will learn how to decompose a large problem into modules and apply the concepts learned in class to solve programming assignments.

You will convey technical knowledge in a clear and concise manner by presenting your work in class to develop skills in oral and written communication.
Using the basics of the Object Oriented Paradigm you will learn how it can be used to model a real life project such as a Point of Sale application, a calculator or phone book application. You will be exposed to solving software problems using application programming interfaces (API) in Java. You will also be exposed to the fundamentals of processing in Java.

**Number of Course Credits**

3 Credits

**Pre-requisites**

**CSCI.UA.002**
Students with extensive programming experience would need to take a test-out exam to take CSCI.UA.0102

Familiarity with basic concepts of programming in some programming language is required (variables, expressions, assignment statements, control statements and basic input output).

**Instructor**

**Anasse Bari, Ph.D.**
Email: abari@nyu.edu
Office: 425 WWH
Phone: 8-3227
Office Hours: (see [http://cs.nyu.edu/~abari/](http://cs.nyu.edu/~abari/))

**Textbook**

**Title:** Introduction to Java Programming, Brief Version
**Edition:** 10th edition
**Note:** You may not need the comprehensive edition, although it might come in as a handy reference if you are planning to take the CSCI 102 course.
**Author:** Y. Daniel Liang
**ISBN-10:** 0-13-359220-0, **ISBN-13:** 978-0-13-359220-7
Topics

The course consists of two parts:

Part 1: Fundamentals of Java Programming

Introduction to computer Science, computer programming and Java
Introduction to Integrated Development Environment: Eclipse
Strings, words and numbers in Java
Primitive data types and expression in Java
Conditional Statements
Iterative Statements
Methods
Arrays and Two-dimensional arrays

Part 2: Object Oriented Programming and Design

Introduction to the object oriented programming paradigm
Objects and Classes
Advanced Object Oriented Programming Features
  Inheritance and Polymorphism
  Graphics and Animations (using Processing)
  Exception Handling and Text I/O
Abstract Classes and Interfaces
Recursion (if time permits)
Introduction to sorting algorithms
ArrayLists in Java

Assignments

The homework assignments will be sent out via the course website and NYUclasses. Below is an overview of the learning outcomes from each homework.

Homework#1: Getting Started with Eclipse and Java

Learning Objectives:

- Getting familiar with Eclipse
- Writing simple programs in Java
- Learning the Scanner class in Java
- Learning basic programming concepts
Homework#2: Expressions, assignment statements and arithmetic operators in Java

Learning Objectives:

- Understanding Java expressions and assignment statements
- Learning how to work with variables in Java
- Learning the process of going from simple algorithms to implementation
- Practicing selection statements and Java’s math library
- Learning how to research for appropriate Java classes and use them properly.
- Learning random number generators in Java

Homework 3: Practicing control statements and Strings.

Learning Objectives:

- Leveraging conditional and iterative statement to implement an ATM Machine
- Using the class String and methods defined in the String class to solve problems
- Learning ASCII code of characters

Homework#4: Practicing modularization and learning methods in Java.

Learning Objectives:

- Getting familiar with Eclipse
- Writing simple programs in Java
- Learning the Scanner class in Java
- Acquiring the ability to explain basic programming related concepts in your own words

Homework#5: Phonebook Java Application using arrays and array lists

Learning objectives:

- Applying the Object Oriented Programming paradigm to design a Phone Book Directory in Java.
- Learning array and array lists in Java
- Implementing sorting algorithms (Optional)
Homework#6: Project with Possessing in Java

Learning objectives:

- Learning advanced OOP concepts using processing
- Developing a graphical user interface like application using processing’s API

Grading Policy

Quizzes

There will be a number of quizzes in this class. 80% of the quizzes will be announced with a one session notice. For instance, you will be notified of a Thursday quiz on Tuesday. It is your responsibility to keep up with class announcements. Pop-up quizzes will be unannounced. Online quizzes will be facilitated through NYUclasses.

In-class Quizzes

There will be no make-up opportunities for in-class quizzes unless in case of an excused absence documentation, as indicated above, is provided.

Your grade will be based on:

- Midterm Exam#1 (20%)
- Midterm Exam#2 (25%)
- Quizzes (5%)
- Homework and Programming Assignments (20%)
- Final Exam (30%)
- Class attendance and participation will be added to your overall final grade.

The following scale will be followed when assigning the final grade:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>95-100</td>
</tr>
<tr>
<td>A-</td>
<td>90-95</td>
</tr>
<tr>
<td>B+</td>
<td>87-90</td>
</tr>
<tr>
<td>B</td>
<td>84-87</td>
</tr>
<tr>
<td>B-</td>
<td>80-84</td>
</tr>
<tr>
<td>C+</td>
<td>76-80</td>
</tr>
<tr>
<td>C</td>
<td>72-76</td>
</tr>
<tr>
<td>D</td>
<td>65-72</td>
</tr>
<tr>
<td>F</td>
<td>less than 65</td>
</tr>
</tbody>
</table>