<div align="center">

COURSE SYLLABUS

City College of New York
The City University of New York
Department of Computer Science

</div>

Course Information:

CSc 221 – Software Design Laboratory   **Credits:** 3   **Class Hour:** 3   **Spring 2019**

Instructor Information:

Ayman Zeidan
azeidan@bmcc.cuny.edu

Course Objectives:

This course focuses on the use of object-oriented programming and design to solve a wide range of computer applications. The course combines the practical aspects of programming exercises with more theoretical discussions of application design methodologies as well as several key internals of object-oriented systems exemplified with Java programming language. Encapsulation, inheritance, and polymorphism are stressed. Goals of the course are:

1. To help you develop proficiency in Java.
2. To help you develop an understanding of object-oriented programming: what it is, why it is important, and how to do it.
3. To extend your command of what is meant by program design.
4. To introduce you to the Java approach to compilation and execution, which involves the Java Virtual Machine executing bytecode.
5. To build a foundation understanding of Design Patterns.

We will study graphical user interfaces (GUIs) in some detail, although it will be impossible to cover more than a fraction of the power that the Java-based graphics packages provide. You will get enough experience with GUI development to learn the rest of that subject on your own; we need course time for things that are not so easily learned by oneself.

Textbook and References:

Title:	Java How to Program (early objects), 11/E
Author:	Paul Deitel & Harvey Deitel
Publisher:	Prentice Hall, 2015
ISBN:	*ISBN-13: 9780133807806* *ISBN-10: 0133807800*

Lecture slide: Will adopt a power-point slide set produced by the textbook author for my class lectures. Other materials will be placed online.

Specific course information:

- Accelerated introduction to Java programming language and its standard library usage. The course covers coding principles, graphic user interface, event-driven programming, design patterns, security issues, and network and mobile computing capabilities. This course also introduces application development under the Android mobile operating system. A small-scale, team-based application development including software specifications and unit and user testing is required.
- Prerequisites: CSc 21200 and Engle 21007 or Engle 21002 Engle 21003

Specific goals for the course and Relationship to student outcomes

	1	2	3	4	5	6
a. The student acquires proficiency in Java to implement and test various small-scale applications in Java programming language	R	R				R
b. The student acquires an understanding of event-driven programming and ability to clearly explain what it is, why it is important, and how to do it	R					R
c. The student is able to list the main principles of software specifications, program design, and unit and user testing	R					R
d. The student acquires a foundation understanding of Java-approach for platform independence and ability to use IDE-based compilation and debugging		R				R
e. The student acquires a foundation understanding of design patterns and ability to use it for project design			I	I	I	
f. The student develops an understanding of legal and security issues related to network-deployed application developments				P		I

I - introductory-level; R - reinforced-level; P - program-level

Assignments and Grading:

The syllabus below shows a tentative timetable for a schedule. There will be six programming assignments distributed every week (counted 60% of your final grade). There will be three in-class, closed-book short exams (15%). Each will cover lecture and reading material since the previous exam. Dates of these exams will be announced beforehand. There is one small-scale programing project realized in two phases (25%). There will be no final exam.

Exam 1	5%
Exam 2	5%
Exam 3	5%
Programming Assignments	60%
Programming Project	25%
	100%

Policies:

- The course work must be carried out individually. Instructions to submit each assignment will be given later. Weekly programming work needs to be submitted on time.
- Warning: the assignment and project work must be done independently. Sharing materials with classmate, especially programming work including logic, and/or modifying the materials to fabricate and reproduce other versions is very seriously treated based on the CUNY Academic Integrity Policy. Do not underestimate my reaction resulted from breaking this rule.
- Late assignment submission will be allowed only within two days. Submission due Monday can be submitted Wednesday of the same week but will incur score deduction of 25%. Submissions after this late due are not allowed.
- Three exams will be given in class. You are expected to arrive at the beginning of the class period, and no extra time will be given for late arrivals. No make-up exams will be considered for absent students.
- Any disagreement resulted from my grading must be reported *in writing* for the resolution.

Communications: I would like the course to run smoothly and enjoyably. Feel free to let me know what you find good and interesting about the course. Let me know sooner about the reverse.

Brief list of topics to be covered (tentative timetable)

Week	Topics	Required Chapters
1	▪ Introduction to Computer and Java; Java Applications; Classes, Objects, Methods and Strings ▪ Assignment #1	Ch.1 – skim Ch. 2 and 3
2	▪ Methods and Strings (cont.); Control Statements; Arrays and ArrayLists	Ch. 4 and 5
3	▪ Methods: A Deeper Look and Arrays and ArrayLists ▪ Assignment #2	Ch. 6 and 7
4	▪ Classes and Objects: A Deeper Look; Inheritance	Ch. 8 and 9
5	▪ 1st Exam ▪ Polymorphism and Interfaces ▪ Assignment #3	 Ch. 10
6	▪ Exception Handling; Android Application Development Basics ▪ Project – Phase I	Ch. 11
7	▪ GUI Components ▪ Assignment #4	Ch. 12
8	▪ GUI Components (cont.); Graphics and Java 2D	Ch. 13
9	▪ Strings, Characters and Regular Expressions; File, Streams and Object Serialization ▪ Assignment #5	Ch. 14 and 15
10	▪ 2nd Exam ▪ Java Security; Secure Coding Guidelines	
11	▪ Android Programming Tutorial; Generic Collections ▪ Project – Phase II	Ch. 20
12	▪ Generic Collections (cont.); Lambdas and Streams	Ch. 17
13	▪ Generic Classes and Methods; Custom Generic Data Structures ▪ Assignment #6	Ch. 21 and 22
14	▪ GUI Components: Part 2; Concurrency	Ch. 23
15	▪ Course Wrap-up; Final project demos	