CS 023: Data Structures & Object Oriented Programming

Term: 2020 Summer Session **Instructor: Staff** **Language of Instruction: English** **Classroom: TBA** **Office Hours: TBA**
Class Sessions Per Week: 5 **Total Weeks: 5** **Total Class Sessions: 25** **Class Session Length (minutes): 120** **Credit Hours: 4**

Course Description:

This course is designed to help students to learn and implement data structures in the software development and algorithm analysis using an object-oriented programming language. Topics around basic and intermediate OOP, GUI, data structure and algorithms analysis will be covered.

Learning Objectives:

Upon completion of this course, students are expected to:

1.   have a good understanding on the basic idea of software development and algorithms;

2.   be familiarized with concepts of data structure and abstract data types;

3.   demonstrate ability to design, analyze and implement data structure and algorithm analysis with an object-oriented programming language;

Course Materials:

Java Foundations: Introduction to Program Design and Data Structures, 4th Edition by John Lewis (Author), Peter DePasquale (Author), Joe Chase (Author)

Publisher: Pearson; 4 edition (February 20, 2016)

Language: English

ISBN-10: 0134285433

ISBN-13: 978-0134285436

Course Format and Requirements:

The course will take place in a computer lab and the course format including lecture, programming project, and in-class discussion.

The specific topics that will be covered in the classes are listed in the course syllabus. The class period will consist of an active learning environment. During a majority of the class time, students will be actively working on problems under the instructor's guides.

Attendance:

Attendance will not be taken but is strongly recommended. Each student will have three allowed absences and no grade deduction will be made for the first three absences. More than three unexcused absences will result in an automatic reduction in your participation grade, for instance from A- to B+. Your active participation in the class is expected and encouraged.

Grading Scale:

A+: 98%-100%
A: 93%-97%
A-: 90%-92%
B+: 88%-89%
B: 83%-87%
B-: 80%-82%
C+: 78%-79%
C: 73%-77%
C-: 70%-72%
D+: 68%-69%
D: 63%-67%
D-: 60%-62%
F: Below 60%

Course Assignments:

Quizzes:

There will be 6 unannounced quizzes through this semester. Each quiz will be on the material covered in previous chapters. All of the quizzes will be closed book and the lowest score will be dropped in final grading. No make-up quizzes will be given. The quizzes will account for 15%of your final grade.

4 Programming Projects

There will be 4 hands-on programming projects based on course need. These programming projects are all individual work. Students are encouraged to exchange and discuss programming knowledge and ideas together. But each student shall submit completed individual work.

The projects aim to enrich students' knowledge on application of learned ideas and concepts on Java, data structure, algorithms by using object-oriented programming language and logic.

The score will be given based on the correctness of the program. It will count for 30% of your grade for the course.

Exams (One Midterm Exam + Final Exam)

Both exams will be based on the knowledge covered in class. Exam questions will includes multiple choice questions and True or false questions, which test your understanding on Java language, basic data structure and object oriented programming principles. It will also includes some short answer or programming question.

Note that the final is cumulative and it will not be taken during the normal class times. Exact time and location for final will be announced later.

No excuse will be accepted if students do not have legitimate excuses for absence. Physician Statement is required for missing the exam due

Course Assessment:

Quizzes	15%
4 Programming Projects	30%
Midterm Exam	25%
Final Exam	30%

Total	100%

Academic Integrity:

Students are encouraged to study together, and to discuss lecture topics with one another, but all other work should be completed independently.

Students are expected to adhere to the standards of academic honesty and integrity that are described in the Shanghai Normal University's *Academic Conduct Code*. Any work suspected of violating the standards of the *Academic Conduct Code* will be reported to the Dean's Office. Penalties for violating the *Academic Conduct Code* may include dismissal from the program. All students have an individual responsibility to know and understand the provisions of the *Academic Conduct Code*.

Special Needs or Assistance:

Please contact the Administrative Office immediately if you have a learning disability, a medical issue, or any other type of problem that prevents professors from seeing you have learned the course material. Our goal is to help you learn, not to penalize you for issues which mask your learning.

Course Schedule:

Week	Topics	Activities
Week 1 (class 1-5)	Go through syllabus Course overview + Introduction **Review of Java Programming:** Editors, Compilers, and Interpreters Development Environments Syntax and Semantics Variables Data Types Expressions Conditionals and Loops **OOP**	▪ Quiz 1 ▪ Quiz 2

	Classes: Creating Objects; Packages; String; Math	
Week 2 (class 6-10)	**OOP** Writing Classes: Members, Relationships, Method Inheritance: Subclasses; The Object Class; Abstract Classes; Polymorphism via Interfaces **GUI** GUI Elements: frames, panels, buttons, action events More GUI Elements (Text Fields, Check Boxes, Radio Buttons, Timers...)	▪ Quiz 3 ▪ Programming project 1
Week 3 (class 11-15)	**GUI** Layout; Mouse and Key events; GUI Design **Data Structures** Arrays: array elements, declaring and using arrays, arrays of objects, 2D arrays Stacks: Stack Collection, Stack ADT, The Array Stack Class, Linked Lists, The Linked Stack Class Queues: A Queue ADT, Code Keys, Queues with links, Queues with arrays	▪ Quiz 4 ▪ Programming project 2 ▪ Midterm Exam
Week 4 (class 16-20)	**Data Structures** Heaps: Heap Implementation, Heap Sort, Priority Queues Trees: Tree Classifications, Tree Traversal, Binary Search Tree, Balanced Binary Search Trees and Implementation	▪ Quiz 5 ▪ Programming project 3

	Exceptions Exception Handling Uncaught Exceptions	
Week 5 (class 21-25)	**Exceptions** Exception Propagation I/O Exceptions **Algorithms** Recursion: Recursion in Math; Recursion vs. Iteration; Direct vs. Indirect Recursion Searching and Sorting Analysis of Algorithms: Algorithm Efficiency; Growth Functions; Big-O Notation Course Summary	▪ Quiz 6 ▪ Programming project 4 ▪ Final Exam