

Course Outline

Department of Computing Science
Faculty of Science

**COMP 1230 – 3
Computer Programming 2
Semester: Fall 2015**

Instructor: TBA
Office: TBA
Office Hours: TBA

Phone/Voice Mail: TBA
E-Mail: TBA

Calendar Description:

This course is a continuation of Computer Programming 1 course. Students are introduced to the foundation for further studies in computer science. Students continue to learn the disciplined approach to the design, coding, and testing of programs in the object oriented paradigm. Students learn object-oriented programming in detail, and are introduced to the data structures and algorithm analysis.

Educational Objectives/Outcomes:

Upon successful completion of the course, the student will demonstrate the ability to:

1. Develop classes and describe how to declare a class
2. Familiar with object-oriented features, such as encapsulation, inheritance, and polymorphism.
3. Understand how to use built-in Java library for sorting and searching.
4. Understand error-handling techniques using exception handling in Java
5. Create an event-driven graphical user interface (GUI) using Swing components: panels, buttons, labels, text fields, and text areas
6. Understand how to use graphics library and packages in Java.
7. Implement input/output (I/O) functionality to read from and write to data and text files and understand advanced concepts of input/output streams.
8. Perform multiple operations on database tables, including creating, reading, updating and deleting using both Java Database Connectivity.
9. Understand concept of multi-threaded programs in Java

Prerequisites:

COMP 1130: Computer Programming 1 or equivalent.

Textbook:

1. Lewis, DePasquale and Chase; Java Foundations, An Introduction to Program Design and Data Structures, latest edition, Pearson Education Inc., 2011, ISBN-13: 978-0-13-337046-1 or ISBN-10: 0-13-337046-1
An e-book/e-chapters are also available at
http://www.coursesmart.com/IR/2170045/9780133449624?__hdv=6.8

Course Topics:

Topic	Specific Objectives	Week #	Book Ch
OOP concepts: Inheritance	Discussion of Inheritance and method overriding.	1 & 2	Ch 8
OOP Concepts: Abstraction and Polymorphism	Discussion of Object Oriented Programming Concepts including abstraction using class and interface, Polymorphism, methods overloading, etc.	3 & 4	Ch 9
Exception Handling and I/O	Implementing error-handling techniques using exception handling in Java and Implementing input/output (I/O) functionality to read from and write to data and text files and understand advanced I/O streams.	5	Ch 10
Sorting and Searching	Example of built-in library for sorting and searching will be shown; Mainly, how to use the built-in sorting and searching library in Java will be discussed without explaining the actual implementation of the algorithms.	6	Instructor Note
Multi-threaded Programming in Java	An introduction to the multithreaded programming in Java will be discussed; some specific topics such as how to create threads, inter threads communication, wait, notify, sleeping of thread, etc will be discussed.	7	Ch 11

Java Graphical User Interface	Introduction to event-driven graphical user interface (GUI) using Swing components: panels, buttons, labels, text fields, and text areas, etc.	8 & 9	Ch 6
Java Graphics	Introduction to simple graphics, coordinate system, drawing basic shapes	10	Appendix "F" & Instructor's Notes
Java Applet	Introduction to Java Applets.	11	Appendix "G" & Instructor's Notes
Introduction to Java Database API: JDBC	Examples and discussion on performing operations on database tables, including creating, reading, updating and deleting using JDBC.	12	Ch 25
Introduction to Java Package/API	A brief introduction to Java packaging technologies – creating, and importing your own package, JAR files, etc.	13	Instructor's Note

Lab/Seminar Topics:

Topic	Specific Objectives	Lab
Exercise on Arrays and Strings	Small programming projects involving use of multi-dimensional array, strings, etc.	1
Exercise on GUI	Small projects involving event-driven graphical user interface (GUI) using Swing components: panels, buttons, labels, text fields, and text areas, etc.	2 & 3
Exercise on OOP concepts: Inheritance, Abstraction and Polymorphism	Small projects to demonstrate the use of inheritance in Java.	4 & 5
	Small projects to demonstrate the use of abstraction using class and interface, methods overloading, method overriding, etc.	6 & 7

Exercise on Exception Handling and I/O operations in Java	Programming assignments to demonstrate exception and I/O functionality. Example, reading/writing data/text from/to files.	8
Exercise on Multithreaded Programming in Java	A simple multi-threaded program to demonstrate how to create thread, inter-thread communication, and use of sleep, notify, wait and other thread related operations in Java.	9
Exercise on Sorting and Searching Library Methods in Java	Simple programs to show how use/call built-in library for sorting and searing in Java.	10
Exercise on Java Applet	Demonstrate how to create an applet in Java, life-cycle of an applet, etc. with a simple example.	11
Exercise on Java Database API: JDBC	A programming exercise to show how to perform operations on database tables, including creating, reading, updating and deleting using JDBC.	12
Exercise on Java Package	How to create own library, how to create and use JAR files, and importing own library into another program	13

ACM / IEEE Course Outline Format

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	Total Hours of Coverage
Programming Languages (PL)	24
Software Development Fundamentals (SDF)	13
Algorithms and Complexity (AL)	2
Discrete Structures (DS)	1
Human-Computer Interaction (HCI)	1

Body of Knowledge coverage

KA	Knowledge Unit	Topics Covered	Hours
AL	Fundamental Data Structure and Algorithms	Simple numerical algorithms Sequential and binary search algorithms	2
PL	Object-Oriented Programming	All-core topics (Tire-1)	7
PL	Functional Programming	Some advanced topics (Tire-2)	5
HCI	Programming Interactive Systems	Model-view controller Event management and user interaction	2
SD F	Algorithm and Design	The concept and properties of algorithms (informal comparison of algorithm efficiency) Iterative and Recursive traversal of data structure Fundamental design concepts and principles (abstraction, program decomposition, encapsulation and information hiding, separation of behavior and implementation)	6
SD F	Development Methods	Program comprehension, program correctness, Simple refactoring, debugging strategies, documentation and programming style.	4
PL	Functional Programming	Some advanced topics such as effect-free programming	4
SF	Cross-Layer Communication	Programming abstraction, interfaces, use of library	3
PL	Advanced Programming Constructs	Exception Handling	4
PL	Event-Driven and Reactive Programming	Introduction to event-driven programming concepts and reactive programming concepts with Java GUI	4