

STANDARDIZED COURSE OUTLINE

SECTION I

SUBJECT AREA AND COURSE NUMBER: CSC 220

COURSE TITLE: Object Oriented Programming Using Java

COURSE CATALOG DESCRIPTION:

This course centers on programming in Java. Java is a complete object oriented programming language derived from C++. Java allows the developer to create executable code that runs in the browser's memory space. From within a Java program you can draw, paint bitmaps, get user events and respond to them, and load URLs. In this course, Java development will be in Microsoft's Visual J++ IDE. An IDE is an integrated development environment, which gives the user one place to do coding, running and debugging. Object oriented programming (OOP) will also be explored. *Formerly listed as CIS 240, not open to students who have successfully completed CIS 240.*

LECTURE HOURS PER WEEK: 3

CREDIT HOURS: 3

LAB HOURS PER WEEK (if applicable): n/a

PREREQUISITE(S): CSC*101 with at least one high-level programming language such as Visual Basic, Cobol, C, C++ or Java Scripting

SECTION II

A. SCOPE:

This course will focus on the following topics: Introduction to software development; Java classes, objects, and events; Java syntax and style; data types, variables, and arithmetic; if-else and Boolean expressions; methods, constructors, and fields; strings; arrays; iterative statements; searching and sorting; files and streams; graphics; GUI components and events; mouse, keyboard, sounds, and images; and OOP concepts and design.

B. REQUIRED WORK:

Will vary by instructor. Students will be expected to do all required readings, assignments, tests, and quizzes as outlined by their instructor.

C. ATTENDANCE AND PARTICIPATION:

Regular attendance, assignment submission timeliness, promptness and class/lab participation will be expected. Instructors will include specific attendance and participation policies requirements in their class syllabi.

D. METHODS OF INSTRUCTION:

Methods may include any of the following: lecture, lecture/discussion, small group, collaborative learning, experimental/exploration, distance learning, student presentations, computer demonstrations, or use of technologies such as audio-visual materials, and computer laboratory equipment. Emphasis will be on hands-on computer exercises and problems.

E. OBJECTIVES, OUTCOMES, and ASSESSMENT

Students' grades will be based on achievement of learning the objectives and outcomes listed below as measured by the instructor's methods of assessment:

LEARNING OBJECTIVES	LEARNING OUTCOMES	ASSESSMENT METHODS
To demonstrate an understanding of:	Student will:	As measured by:
Introduction to software development	a) Use algorithms and flowcharts to write a program b) Use a Java compiler c) Apply basic OOP concepts	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
Java classes, objects, and events	a) Implement source files, instances, fields, and methods in code	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
Java syntax and style	a) Implement comments and reserved words in code b) Explain Java naming conventions	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
Data types, variables, and arithmetic	a) Implement source fields, local variables and constants in code b) Apply the concept of "scope" in program code	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
If-else and Boolean expressions	a) Implement basic if-else and nested if-else statements in code b) Use switch statements to replace nested if-else statements c) Use relational operators in conditions	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
Methods, constructors, and fields	a) Explain and apply the concepts of "overloaded", "public", "private", "static", and "non-static" in code	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations

Strings	<p>a) Implement literals, methods, and immutability in code</p> <p>b) Convert data from one data type to another</p>	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
Arrays	<p>a) Implement both one and two dimensional arrays in code</p> <p>b) Access array elements</p>	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
Iterative statements	<p>a) Implement while loops, do-while loops, for loops, and break statements in code</p>	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
Searching and sorting	<p>a) Explain sequential and binary searches, selection, insertion, and merge sorts</p> <p>b) Implement a basic sort in code</p>	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
Files and streams	<p>a) Explain and apply the concepts of “consoles I/O” and “file I/O” in code</p>	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
Graphics	<p>a) Use coordinates, shapes and placed text in Java</p> <p>b) Write a program using graphical ideas</p>	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
GUI components and events	<p>a) Implement swing, Jpanel, and JFrame in code</p>	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations
Mouse, keyboard, sounds, and images,	<p>a) Use a basic interface to capture input from a mouse or keyboard</p> <p>b) Use images and sounds to enhance an interface</p>	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations

		Presentations
OOP concepts and design	a) Explain the “inheritance”, “encapsulation”, and “polymorphism” OOP concepts b) Use advanced concepts to write a more advanced object-oriented program	<ul style="list-style-type: none"> • Homework/Lab assignments; • Written and Oral activities; • Quizzes and Exams; • Online Computer Exercises; • Programming Projects and Presentations

F. TEXT(S) AND MATERIALS:

An appropriate Object Oriented Programming Using Java Text, such as:

Text: Java Software Solutions (*current edition*)

Author: Lewis

Publisher: Addison Wesley

G. INFORMATION TECHNOLOGY:

This course is an information technology course and will require extensive computer lab time both for teaching and performing assignments. Students will require network accounts with access to Internet Explorer and a Java compiler as well as file storage space.