

## COURSE DESCRIPTION

Department and Course Number	<b>CS 201</b>	Course Coordinator	<b>Johnstone</b>
Course Title	<b>Introduction to Object-Oriented Programming</b>	Total Credits	<b>4</b>

### Current Catalog Description

*Fundamental concepts of object oriented programming. Syntax and semantics of Java, an object oriented programming language. Principles of program design and algorithm development strategies. Classes, abstract data types, arrays, flow control, functions, overloading, exception handling, debugging, I/O, applets. This class has a laboratory component.*

### Textbook

*Java Software and Solutions: Foundations of Program Design, 5<sup>th</sup> ed., by John Lewis and William Loftus, Addison-Wesley, 2007.*

References *None*

### Course Goals

*This course is designed to introduce the student to the fundamental concepts of object-oriented programming (OOP). There is an extensive problem solving and programming component using the Java programming language. This course requires many hours of programming outside of class.*

Prerequisites by Topic *Intermediate Algebra with grade of C or better*

### Major Topics Covered in the Course

*Fundamental concepts of object oriented programming. Syntax and semantics of Java, an object oriented programming language. Principles of program design and algorithm development strategies. Classes, abstract data types, arrays, flow control, functions, overloading, exception handling, debugging, I/O, applets.*

Laboratory projects (specify number of weeks on each)

- 1. Input and compute the area and circumference of circles (1)*
- 2. Develop an applet containing random shapes (1)*
- 3. Evaluate two-dimensional arrays for "magic square" characteristics (1)*
- 4. Implement inheritance in classes representing individuals in a university (1)*
- 5. Include exception handlers to catch run-time errors (1)*
- 6. Design Robocode robots for competition at the end of the term (4)*

## Estimate CSAB Category Content

	CORE	ADVANCED		CORE	ADVANCED
	7			2	
Data Structures	<hr/>	<hr/>	Computer Organization and Architecture	<hr/>	<hr/>
	15			20	
Algorithms			Concepts of Programming		
Software Design	<hr/>	<hr/>	Languages	<hr/>	<hr/>

## Oral and Written Communications

*None*

## Social and Ethical Issues

*None*

## Theoretical Content

*Algorithm development (4 hours)*

## Problem Analysis

*The students learn to analyze different approaches to programming problems and to think logically about iterative processes. Tracing of programs and debugging teach the student to simulate a solution like a computer.*

## Solution Design

*The student learns to divide a problem into parts using functions, and to build the interface between the calling and called functions, such as formal parameters and scope. The student is exposed to elementary data structures such as arrays and the ideas of data abstraction. Flow control must be used to develop efficient and elegant solutions.*