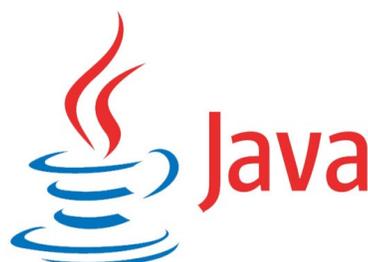


CS 230 Intermediate Programming Using Java Spring 2014

Class Number: 20092 (4 credit units)
Room: 1705
Lecture/Lab TTh 5:20-8:20pm

Instructor: Rick Castrapel
Phone: (760) 355-6505
Office: Room 2773
Email: rick.castrapel@imperial.edu
Web Site: <http://www.imperial.edu/rick.castrapel>
Textbook: Introduction to Java Programming Brief Version
 9th Ed., Liang, Pearson/Prentice Hall 2013
Office Hours: MW 5:30-6:30pm, TR 2:00-3:00pm
 or by appointment



Description:	This course provides an intermediate treatment of object-oriented programming using the Java programming language. Topics included (but not limited to) in this course are abstract classes, inheritance, polymorphism, interfaces, graphical user interfaces (GUI), graphics, event-driven programming, exception handling, file input, file output, and applets. Emphasis is on learning through hands-on programming exercises.
Student Learning Outcomes:	<ol style="list-style-type: none"> 1. Understand Objects and Classes, and their implementations in the Java language. 2. Understand Strings and Text IO, and their implementations in the Java language. 3. Understand Inheritance, Polymorphism, Abstract Classes and Interfaces, and their implementations in the Java language. 4. Write a complete Java program that declares listener classes and inner classes, registers listener objects, and correctly deals with ActionEvent, MouseEvent, and KeyEvent.
Prerequisites:	CS 220 – Beginning Object Oriented Programming in Java, Math 90
Cell Phones:	Keep cell phones turned off during class and labs.
Blackboard:	http://www.imperial.blackboard.com Please use the first part of your IVC Email Address in the username field. For the password field, please use your WebSTAR/Student Portal PIN.
Keeping Up:	Don't let yourself fall behind. If you feel you are slipping, SEE ME . This is urgent . It is my goal and that of the Imperial Valley College Science Division that you succeed.
Lab:	Classes and labs will take place in the Computer Science Lab room 1705.
Dropping:	You may be dropped from this class if you miss the first day or if you miss three or more class sessions total. The last day to drop this class is April 11. After that date, I must give you a letter grade. It is your responsibility to drop, not mine.
DSP&S:	Any student with a documented disability who may need educational accommodations should notify the Disabled Student Programs and Services (DSP&S) office as soon as possible. Room 2117 Health Sciences Building (760) 355-6312.
Grading:	There will be 1 midterm test, worth 100 points. There will be a comprehensive final exam worth 100 points. Programming assignments will be given throughout the semester and are worth 200 points total. Programming assignments will be graded 20% on style and documentation and 80% on correctness.

Grading Policy

Midterm Test	100 points
Programming Assignments	200 points
Final Exam	100 points
Total	400 points

Grading Scale

90-100 %	A
80-89%	B
70-79%	C
60-69 %	D
< 60 %	F

Academic Integrity is assumed and necessary. You will be treated as an adult professional and will be expected to behave accordingly. You must follow the Computer Sciences Code of Conduct. Programming assignments must be done individually. Failure to do so will result in a violation of the Academic Honor Code. The following cases will be considered as violations: identical code, and extremely similar code. Violations will be reported to the Office of Vice President of Student Services. Disruptive students will be required to leave the class for the day. Continued disruptive behavior, cheating, plagiarism or deliberate unsafe computer use may result in severe academic penalty. See the college bulletin for details.

CS 230 Spring 2014 Tentative Schedule

Date	Text	Event	Topic
01/21/14	Ch 8	Course Introduction	Objects and Classes
01/23/14			
01/28/14			
01/30/14			
02/04/14	Ch 9		Strings and Text I/O
02/06/14			
02/11/14			
02/13/14	Ch 10		Thinking in Objects
02/18/14			
02/20/14	Ch 11		Inheritance and Polymorphism
02/25/14			
02/27/14	Ch 15		Abstract Classes and Interfaces
03/04/14			
03/06/14		Midterm Exam	Chapters 8-11,14
03/11/14	Ch 20		Recursion
03/13/14	Handouts		Linked Lists
03/18/14	Handouts		Binary Trees
03/20/14	Ch 12		GUI Basics
03/25/14			
03/27/14	Ch 13		Graphics
04/01/14			
04/03/14	Ch 14		Exception Handling and Text I/O
04/08/14			
04/10/14	Ch 16		Event-Driven Programming
04/15/14			
04/17/14	Ch 17		GUI Components
04/22/14			
04/24/14			
04/29/14	Ch 17		GUI Components
05/01/14	Ch 19		Binary I/O
05/06/14			
05/08/14	Ch 18		Applets and Multimedia
05/13/14			
05/15/14		Final Exam	Comprehensive Final

CS230 Students Learning Outcomes and Course Objectives

Upon course completion, the successful student will have acquired new skills, knowledge, and or attitudes as demonstrated by being able to:

1. Understand Objects and Classes, and their implementations in the Java language.(ILO1, ILO2, ILO4)
2. Understand Strings and Text IO, and their implementations in the Java language.(ILO1, ILO2, ILO4)
3. Understand Inheritance, Polymorphism, Abstract Classes and Interfaces, and their implementations in the Java language.(ILO1, ILO2, ILO4)
4. Write a complete Java program that declares listener classes and inner classes, registers listener objects, and correctly deals with ActionEvent, MouseEvent, and KeyEvent.(ILO1, ILO2, ILO4)

Upon satisfactory completion of the course, students will be able to:

1. Design a new class and implement the class as a part of a complete Java program that declares and defines the class, instantiates objects of the class, and invokes instance and static methods of the class to satisfy the program's functional specification.
2. Design, code, and test a complete Java program that performs a simple text -processing task by correctly employing text file input/output and correctly instantiating and manipulating objects of the Character, String and StringBuffer classes and at least one numeric wrapper class.
3. Design, code, and test a complete Java program that correctly employs inheritance and polymorphism, by implementing a class hierarchy of superclasses and subclasses, and polymorphic invocation of their respective methods to satisfy the program's functional specification.
4. Design, code, and test a complete Java program that correctly employs abstract classes, abstract methods, and interfaces to satisfy the program's functional specification.
5. Design systems by identifying the relationships among them and design classes using the class-design guidelines.
6. Design, code and test Java programs utilizing elementary data structures: stacks, queues and binary trees.
7. Write a complete Java program that employs a GUI and graphics to satisfy its functional specification.
8. Write a complete Java program that implements the Graphics class, the paintComponent method, panels as canvases, incorporates strings, lines, rectangles, ovals, arcs, and polygons, and displays images in a GUI component.
9. Write a complete Java program that declares listener classes and inner classes, registers listener objects, and deals with ActionEvent, MouseEvent, and KeyEvent.
10. Create a GUI using components including JButton, JCheckBox, JRadioButton, JLabel, JTextField, JTextArea, JComboBox, JList, JScrollBar, and JSlider.
11. Develop applets using init, start, stop, and destroy methods in the Applet class, embed the applet in Web pages, and run the applets from a viewer and a browser.

Imperial Valley College Computer Sciences Code of Conduct

We believe that everyone has a right to work in an environment where people treat one another honestly and fairly. Because academic dishonesty can threaten this environment we will pursue abuses of the policies outlined below aggressively.

When you submit any piece of work for grading or other evaluation, the reader will assume that you are the sole author of all aspects of it. The expectation is that you are the originator of every idea and author of every sentence in an essay, help file, or other document, that you wrote every line of code, that you designed every data structure and created every piece of data. In practice, you will often have good reason to use other people's work or to collaborate with others in creating a work that you will submit. In these cases, it is your responsibility to make the reader clearly aware of what has come from other sources. If a reasonable reader would assume, on reading your work, that some part was created by you alone when in fact it was created by someone else or by you in partnership with someone else, that reader has been misled. It is your responsibility to prevent such misimpressions, and the department will hold students accountable both for intentionally misleading readers and for failing to prevent reasonable misimpressions.

Code Plagiarism.

Computer science is a discipline where it is difficult to draw a precise line between acceptable and unacceptable collaboration. On the one hand we want to encourage you to try out other peoples' code; code reuse is an area of active research within computer science. On the other hand you will learn to write code only if you do it yourself. You are not learning and have crossed the line of acceptable behavior if you do not understand the solution you have submitted. We have the right to ask students to explain the code they submit. If you have "reused" someone else's code to an extent that you feel a need to change variable names or slightly rearrange the order of statements, then you have also violated the honor code. We also reserve the right to use electronic tools to check code for plagiarism. By submitting code for grading in any computer science course, you grant the instructor a license to send a copy of that code for plagiarism analysis to a research service, such as [MOSS](#). The instructor, or their service, may compare your code against other students' code, or compare their code to yours. Give credit to someone else's ideas with a citation rather than turning in their work as your own.

Text Plagiarism.

When you hand in an essay or other writing assignment, you must give credit to your sources. You must provide a reference for any idea, conclusion, information or data that you got from another source (such as a book, an article on the Net, or a person). If you use someone's words, you must show that you are quoting them (use quotation marks or indent long quotes) and your reference should show your exact source (such as the page number of the article or book). If you quote someone, you must quote them accurately, word for word. To avoid plagiarizing, you might find the following articles useful:

- [Cheating and Plagiarism](#) in Regulations section of IVC General Catalog
- [How Not to Plagiarize](http://www.utoronto.ca/writing/plagsep.html) at <http://www.utoronto.ca/writing/plagsep.html>.
- [Citing Sources and Avoiding Plagiarism](http://www.lib.duke.edu/libguide/cite/works_cited.htm) at http://www.lib.duke.edu/libguide/cite/works_cited.htm.

By submitting a writing assignment for grading in any computer science course, you grant the instructor a license to send a copy of that assignment for plagiarism analysis to a research service, such as [TurnItIn](#). The instructor, or their service, may compare your paper against other students' papers, or compare their papers to yours.

Social Responsibility.

Many people use our machines: students, faculty, staff, and outside visitors. Our machines affect other machines on and off campus and they affect the users of these machines. It is not hard to abuse others by mailing ``spam," ``flaming" to newsgroups, being a ``cracker," displaying digital pornography, bogging down the CPU with processes, or hogging the printer. We expect your use of computer resources will be based on the Golden Rule: do unto others as you would have them do unto you. Poor social responsibility because you are new is one thing, but malicious practices are another matter and will not be tolerated. Do not use BitTorrent or messenger services on IVC Computer Science computers, as these are a common source of computer viruses.

Right to Privacy.

You are encouraged to store *electronic property* on computers provided for your use by Computer Sciences, and you have a privacy right to this information. Others also have a right of privacy to the property they store on our computers. You should not search other's file systems, read their mail, scan or remove their files, try to crack their password, login as someone else, intercept other's network traffic, install viruses, or otherwise violate the right to privacy of others. We will not intentionally abuse your right to privacy. However, to administer our machines we may need to do things you should not, for example, we may need to try to crack your password to verify that it is secure, or kill your processes, or remove your files, or read your email, or otherwise invade your privacy when we suspect you are an abuser of our systems.

Discrimination:

It is the policy of the university that all students, faculty, staff, and guests enjoy an environment free from all forms of discrimination, including ethnic, racial, religious, and sexual harassment.

Disclaimers.

The Imperial Valley College Catalog, and the Student Handbook have additional guidelines on campus standards, behavior, discipline, complaint resolution, etc. The Computer Science Honor Code does not replace or supersede these policies. Faculty teaching computer science courses may establish other *honor* criteria for their classes.

As our machines are part of a larger international network, we assume certain responsibilities as a member of a growing electronic community. Exercising this responsibility may require us to search for suspected abusers of our or others computers. If you suspect that someone has violated your rights as a user of our machines, inform the systems administrator; do not attempt to track them down yourself.

Ideas for this code of honor have been collected from other universities, most notably, Stanford University and the University of Florida.