1. OO Design and principles:
    a. Introduce a simplified OO analysis and design methodology
    b. Present the concepts of design patterns and choose the appropriate design patterns to follow to meet your application goals
    c. Introduce core UML concepts
    d. Present the concept of a software framework
    e. Properly document the software system
2. Java Language:
    a. Make students proficient in the use and creation of interfaces and inheritance hierarchies
    b. Make students proficient in the Java type system
    c. Introduce threads and thread safety
    d. Introduce Java generics
3. GUI Programming:
    a. Introduce GUI toolkits, including basic widgets, the event handling mechanism, advanced graphics programming and animation.
4. Java reflection:
    a. Introduce the basic concepts of reflection programming.
    b. Introduce methods and approaches utilized in Java reflection programming.
5. IO, web, and networking programming:
    a. Introduce methods and approaches available for IO, web, and network programming.

Optional Texts/Readings (no required text)

This class does not require a mandatory textbook. Google is your friend! Always refer to the Java API specification documentation.

Optional textbook (I will not be teaching by it)

Object Oriented Design and Patterns
Author: Cay Horstmann
A newer version of the book is currently under development. Resources can be found at
http://horstmann.com/oodp3/

Other optional readings

Design Patterns in Java | Edition: 2
Author: Steven John Metsker, William C. Wake
ISBN:9780321333025
Publication Date:04/21/2006
Publisher: Addison-Wesley.
Effective Java (Java Series) | Edition: 2
Author: Joshua Bloch
ISBN:9780321356680
Publication Date:05/18/2008
Publisher: Addison-Wesley

Java Concurrency in Practice
Author: Brian Goetz, Tim Peierls, Joshua Bloch
ISBN:9780321349606
Publication Date:05/23/2006
Publisher: Addison-Wesley

Other technology requirements / equipment / material

Java 8 or higher. (This is the minimum required version though most of you may be using Java 9. We will discuss Java 9 as needed in class).
IDE is not required but could be helpful - Eclipse or Netbeans. You can choose to use any text editor to write your programs. IDEs simply makes writing your programs easier as IDEs usually have additional features that save time and programming effort (e.g. automatic documentation, automatically prepopulating getters and setters, etc.).

Course Requirements and Assignments

SJSU classes are designed such that in order to be successful, it is expected that students will spend a minimum of forty-five hours for each unit of credit (normally three hours per unit per week), including preparing for class, participating in course activities, completing assignments, and so on. More details about student workload can be found in University Policy S12-3 at http://www.sjsu.edu/senate/docs/S12-3.pdf.

- Each student is expected to be present, punctual, and prepared at every scheduled class and lab session. It is assumed that the students already have basic knowledge of digital Boolean logic and fundamentals of programming.

- Attendance is NOT optional though it does not form any part of your grade. Individual participation is also required. There will be no make-ups for missed midterm or assignments, unless any special arrangements is made with the instructor beforehand. The student is responsible for any material he/she may have missed.

- There will be 6-7 homework assignments (some of which might be team based), one final project, one midterm and final exam. All homework should be submitted through Canvas. N p/F3 12 Tf1 0 0 1 36 551.02 8s ry i no r

Incomplete work:

Points will be deducted for incomplete question responses and solutions that are partially functional. Consult

Tentative schedule. Subject to change with notice.
