



The student will develop and demonstrate object oriented programs using java language constructs and manipulate data structures using the collection classes. These programs will exhibit OOP features such as inheritance and polymorphism and explore programming techniques including multithreading, error handling with exceptions and file I/O.

Various java packages will be explored including the Swing and graphics packages as well as packages for accessing databases via java.

The programs written in this course will be primarily console based and Swing gui based applications. Web application programming in java will be introduced in a subsequent course.

## **II.LEARNING OUTCOMES AND ELEMENTS OF THE PERFORMANCE:**

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

- 1.Discuss and apply programming concepts in modern GUI environments with emphasis on event driven systems (instructor supplied notes)

### Potential Elements of the Performance:

- discuss the following and then apply that knowledge to writing a simple application in various environments:
  - implications of event driven systems
  - implications of multi platform computing and the role of Java in multi platform computing

- 2.Become familiar with the Java development environment, in particular the Java SDK in UNIX and Windows, to create and manage programming projects and to become familiar with the documentation system. Development will begin using command line tools but the primary development environment will be the Netbeans IDE.

### Potential Elements of the Performance:

- discuss the history of Java
- describe the Java environment
- write a java applet
- become proficient in the use of on-line resources and documentation systems
- write Java apps using
  - control structures
  - predefined java objects/modules and their methods
  - arrays
  - strings
  - graphics
  - files and streams

- 3.Review object oriented programming concepts and principles and apply those to writing Java program.

Potential Elements of the Performance:

- implement a class
- create and use objects based on user defined classes.
- create and use a package
- demonstrate through example programs
- sub/superclassing.
- inheritance polymorphism
- abstract/concrete classes
- interfaces

4. Write programs incorporating basic and advanced GUI components as well as programs utilizing graphics objects of the Java2D package.

Potential Elements of the Performance:

- demonstrate an understanding of the following concepts and components by writing example programs
- graphics contexts and graphics objects
- manipulate colors and fonts
- graphics methods for drawing lines, rectangles, 3D rectangles, ovals arcs and various other graphic primitives.
- Graphics methods of the [Java2D](#) API to draw lines, rectangles, ellipses, arcs and general paths.
- Specify paint and stroke characteristics of shapes.
- Build GUI interfaces
- understand and implement basic GUI components and incorporate event handling
- utilize layout managers
- create and manipulate menus, popups and windows
- customize JPanel objects
- create applications that run as either applications or applets
- create multiple document interface applications

5. Write programs that demonstrate an understanding of exception handling.

Potential Elements of the Performance:

- Write programs incorporating the following:
- try/catch blocks
- throw exceptions
- use "finally" to release resources

6. Write programs that demonstrate an understanding of multi threading.

Potential Elements of the Performance:

- understand what multi threading is and how it improves performance.
- Create, manage and destroy threads.
- Understand thread synchronization
- understand daemon threads and thread groups

7. Write programs that demonstrate an understanding of files and streams.

- create, read and write files
- use FileInputStream and FileOutputStream classes
- use ObjectInputStream and ObjectOutputStream classes
- use RandomAccessFile class
- use JFileChooser
- use the File class

8. Write programs demonstrating database access.

- understand and apply the relational database model (review)
- use classes and interfaces of java.sql packages to query, create, insert and delete data in a database
- understand and use SQL to perform database queries

### **III. TOPICS:**

1. Introduction to event driven programming and Java.
2. The JAVA programming Environment.
3. Object Oriented Programming.
4. Writing Java Applications utilizing basic and advanced GUI components
5. Writing Java Applications utilizing exceptions
6. Writing Java Applications utilizing multi threading
7. Writing Java Applications demonstrating files and streams
8. Writing Java Applications demonstrating database access

### **IV. REQUIRED RESOURCES/TEXTS/MATERIALS:**

- Text: - Small JAVA How To Program,  
Seventh Edition,  
Deitel & Deitel,  
Pearson/Prentice Hall,  
ISBN 0-13-148660-8
- instructor supplied notes
  - various magazine articles to be supplied

**V. EVALUATION PROCESS/GRADING SYSTEM:**

The mark for this course will be arrived at as follows:

Tests:

outcomes #1- #3 40%

outcomes #4 - #8 30%

Assignments:

outcomes #1 - #3 15%

outcomes #4 - #8 15%

Total 100%

NOTE\*\* A passing grade in both the Test and Lab portion is required to pass this course.

The following semester grades will be assigned to students:

<b>Grade</b>	<b>Definition</b>	<i>Grade Point Equivalent</i>
A+	90 – 100%	
A	80 – 89%	4.00
B	70 - 79%	3.00
C	60 - 69%	2.00
D	50 – 59%	1.00
F (Fail)	49% and below	0.00
CR (Credit)	Credit for diploma requirements has been awarded.	
S	Satisfactory achievement in field /clinical placement or non-graded subject area.	
U	Unsatisfactory achievement in field/clinical placement or non-graded subject area.	
X	A temporary grade limited to situations with extenuating circumstances giving a student additional time to complete the requirements for a course.	
NR	Grade not reported to Registrar's office.	
W	Student has withdrawn from the course without academic penalty.	

## VI. SPECIAL NOTES:

### Disability Services:

If you are a student with a disability (e.g. physical limitations, visual impairments, hearing impairments, or learning disabilities), you are encouraged to discuss required accommodations with your professor and/or the Disability Services office. Visit Room E1101 or call Extension 2703 so that support services can be arranged for you.

### Retention of Course Outlines:

It is the responsibility of the student to retain all course outlines for possible future use in acquiring advanced standing at other postsecondary institutions.

### Communication:

The College considers **WebCT/LMS** as the primary channel of communication for each course. Regularly checking this software platform is critical as it will keep you directly connected with faculty and current course information. Success in this course may be directly related to your willingness to take advantage of the **Learning Management System** communication tool. Faculty reserves the right to use an alternative to LMS.

### Plagiarism:

Students should refer to the definition of “academic dishonesty” in *Student Code of Conduct*. Students who engage in academic dishonesty will receive an automatic failure for that submission and/or such other penalty, up to and including expulsion from the course/program, as may be decided by the professor/dean. In order to protect students from inadvertent plagiarism, to protect the copyright of the material referenced, and to credit the author of the material, it is the policy of the department to employ a documentation format for referencing source material.

### Course Outline Amendments:

The professor reserves the right to change the information contained in this course outline depending on the needs of the learner and the availability of resources.

Substitute course information is available in the Registrar's office.

**Tuition Default:**

Students who have defaulted on the payment of tuition (tuition has not been paid in full, payments were not deferred or payment plan not honoured) as of the first week of *<choose November, March, or June>* will be removed from placement and clinical activities. This may result in loss of mandatory hours or incomplete course work. Sault College will not be responsible for incomplete hours or outcomes that are not achieved or any other academic requirement not met as of the result of tuition default. Students are encouraged to communicate with Financial Services with regard to the status of their tuition prior to this deadline to ensure that their financial status does not interfere with academic progress.

**VII. PRIOR LEARNING ASSESSMENT:**

Students who wish to apply for advance credit transfer (advanced standing) should obtain an Application for Advance Credit from the program coordinator (or the course coordinator regarding a general education transfer request) or academic assistant. Students will be required to provide an unofficial transcript and course outline related to the course in question.

Credit for prior learning will also be given upon successful completion of a challenge exam or portfolio.