

I. COURSE DESCRIPTION:

This course provides an introduction to Java and continues the study of programming languages begun in a previous course.

The student will apply knowledge of program structure and programming constructs such as selection, looping and data structures, to the writing of programs.

In addition the concepts of objects and classes, inheritance and polymorphism will be introduced and applied in the writing of programs.

The course continues with an introduction to GUI programming with an emphasis on event driven programming and concludes with exception handling and binary I/O.

Programs will be written using the Netbeans IDE in the Windows Operating System environment.

II. LEARNING OUTCOMES AND ELEMENTS OF THE PERFORMANCE:

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

1. Write structured code in the Java programming environment.Potential Elements of the Performance:

- Describe the history of the Java programming environment.
- Compare and contrast Java and C++.
- Describe the components of a Java program including attributes and methods.
- Describe the Java programming environment and the process of Java program development/execution.
- Describe primitive Java data types.
- Describe and apply knowledge of data scope.
- Describe and apply various collection constructs such as arrays and lists.
- Write programs with multiple methods that illustrate parameter passing and return of data.
- Write and debug simple Java applications in command line and IDE environments.
- Perform I/O using keyboard, screen and files.
- Apply all of the above in the writing of programs.

2. Utilize the various control structures available with Java.Potential Elements of the Performance:

- Define algorithm.
- Describe the concepts of sequential execution and transfer of control using “if” and “switch” statements.
- List and describe the looping structures available with Java such as “for”, “for each”, “while” and “do while” constructs.
- Write programs utilizing the control structures available with Java.

3. Write Java programs using objects.Potential Elements of the Performance:

- Describe objects and classes and the relationship between them.
- Describe and apply the use of constructors.
- Create objects and access them via object reference variables.
- Differentiate between object reference types and primitive data type variables.
- Describe and apply the public, private and protected visibility modifiers.

4. Inheritance and polymorphism and other OOP constructs.Potential Elements of the Performance:

- Develop a subclass from a superclass through inheritance.
- Invoke super class methods and constructors using the “super” keyword.
- Distinguish and differentiate between the overloading and overriding of methods.
- Understand and apply the concept of polymorphism.
- Describe and apply casting.
- Describe and apply the ArrayList collection class.

5. Abstract Classes and Interfaces.Potential Elements of the Performance:

- Describe and apply abstract classes.
- Describe and apply interfaces.

6. GUI Interfaces and Event Driven Programming

Potential Elements of the Performance:

- Create user interfaces using frames, panels, and Swing widgets.
- Understand and apply layouts.
- Understand event driven programs.
- Understand and apply events, event listeners and event methods.
- Write programs that deal with action events.
- Write programs that deal with mouse events.
- Understand and apply exceptions in the handling of errors.
- Discover how I/O works in the java environment and write programs that read and write data and read and write objects to files.

III. TOPICS:

1. Write structured code in the Java programming environment.
2. Utilize the various control structures available with Java.
3. Write Java programs using objects.
4. Inheritance and polymorphism and other OOP constructs.
5. Abstract Classes and Interfaces.
6. GUI Interfaces and Event Driven Programming

IV. REQUIRED RESOURCES/TEXTS/MATERIALS:

Instructor supplied notes. See

<https://sites.google.com/site/saultcollegeit/courses/csd211-fall-2010/Introduction-to-Java>

V. EVALUATION PROCESS/GRADING SYSTEM:

Theory Tests and Quizzes	60%
<u>Laboratory Work and Tests</u>	<u>40%</u>
Total	100%

NOTE: *It is required to pass both the theory and the assignment part of this course. It is not possible to pass the course if a student has a failing average in the tests and quizzes but is passing the assignment portion, (or vice versa).*

The following semester grades will be assigned to students:

Grade	<u>Definition</u>	<i>Grade Point Equivalent</i>
--------------	--------------------------	-----------------------------------

A+	90 – 100%	4.00
A	80 – 89%	
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:

Absenteeism will affect a student's ability to succeed in this course. Absences due to medical or other unavoidable circumstances should be discussed with the instructor. Students are required to be in class on time and attendance will be taken within the first five minutes of class. A missed class will result in a penalty in your marks unless you have discussed your absence with the professor as described above. The penalty depends on course hours and will be applied as follows:

Course Hours	Deduction
5 hrs/week (75 hrs)	1% / hr
4 hrs/week (60 hrs)	1.5% /hr
3 hrs/week (45 hrs)	2% /hr
2 hrs/week (30 hrs)	3%/hr

Absentee reports will be discussed with each student during regular meetings with Faculty Advisors. Final penalties will be reviewed by the professor and will be at the discretion of the professor.

VII. COURSE OUTLINE ADDENDUM:

The provisions contained in the addendum located on the portal form part of this course outline.