SAULT COLLEGE | 443 NORTHERN AVENUE | SAULT STE. MARIE, ON P6B 4J3, CANADA | 705-759-2554

COURSE OUTLINE

CSD221

Prepared: Fred Carella   Approved: Cory Meunier

Course Code: Title	CSD221: JAVA II
Program Number: Name	2090: COMPUTER PROGRAMMER
Department:	COMPUTER STUDIES
Semester/Term:	18W
Course Description:	This course introduces students to the concepts of Object-Oriented Programming and applies them in practical problem-solving exercises. The course presently uses the Java programming language and the Netbeans IDE as the development environment.This course builds on the skills developed in previous courses, in Java, C++ and Python.
Total Credits:	4
Hours/Week:	4
Total Hours:	60
Prerequisites:	CSD211
This course is a pre-requisite for:	CAR300, CSD309, CSD318, CSD322, CSD325
Vocational Learning Outcomes (VLO's): **Please refer to program web page for a complete listing of program outcomes where applicable.**	**2090 - COMPUTER PROGRAMMER** #2. Develop, test, document, deploy, and maintain secure program code based on specifications. #3. Perform routine maintenance on a database. #4. Apply knowledge of networking concepts to develop, deploy, and maintain program code. #6. Use relevant methodologies, policies, and standards to develop secure program code. #8. Conform to workplace expectations found in information technology (IT) environments.
Essential Employability Skills (EES):	#1. Communicate clearly, concisely and correctly in the written, spoken, and visual form that fulfills the purpose and meets the needs of the audience. #2. Respond to written, spoken, or visual messages in a manner that ensures effective communication. #4. Apply a systematic approach to solve problems. #5. Use a variety of thinking skills to anticipate and solve problems. #6. Locate, select, organize, and document information using appropriate technology and information systems. #9. Interact with others in groups or teams that contribute to effective working relationships and

the achievement of goals.
#10. Manage the use of time and other resources to complete projects.
#11. Take responsibility for ones own actions, decisions, and consequences.

Course Evaluation:

Passing Grade: 50%, D

Other Course Evaluation & Assessment Requirements:

The student must pass both the lab and test portions of the course.

Attendance:
Sault College is committed to student success. There is a direct correlation between academic performance and class attendance, therefore, for the benefit of all its constituents, all students are encouraged to attend all of their scheduled learning and evaluation sessions. This implies arriving on time and remaining for the duration of the scheduled session.

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.

Grade
Definition Grade Point Equivalent
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.

Evaluation Process and Grading System:

Evaluation Type	Evaluation Weight
Labs	40%
Tests	60%

Books and Required Resources:

Instructor supplied resources
Resources, texts and materials will be supplied by your professor. Details will be provided in class.

Course Outcomes and Learning Objectives:

Course Outcome 1.

Define, describe and implement the various foundational elements of an object oriented system.

Learning Objectives 1.

- Define and describe “classes”.
- Define and describe “objects”.
- Define and describe the various components of a class including attributes, accessors and mutators.
- Define the relationship between classes and objects.
- Create objects based on classes.
- Define variables of various data types.
- Define and implement programs that demonstrate variable scopes including static, local and class scope.
- Define and call methods, with and without parameters.
- Write and debug programs that demonstrate all of the above.

Course Outcome 2.

Demonstrate an understanding of the components of an object oriented program.

Learning Objectives 2.

- Write programs comprised of various objects and have those objects interact.
- Demonstrate knowledge of and implement data using the various collection classes including lists and sets.
- Demonstrate knowledge of and implement programs using various libraries.
- Read class documentation.
- Write class documentation.
- Demonstrate an understanding of the package system and the structure of a project.
- Test and debug programs using various methodologies such as unit testing and regression testing.

Course Outcome 3.

Demonstrate an understanding of and implement the concepts of class design

Learning Objectives 3.

- Define and be able to demonstrate

o Abstraction
o Encapsulation
o Generalization
o Polymorphism
o Responsibility driven design.
o Coupling
o Cohesion
o Refactoring

Course Outcome 4.

Define and write programs that demonstrate inheritance and polymorphism.

Learning Objectives 4.

- Define the benefits of inheritance and polymorphism and write programs that demonstrate each.
- Use inheritance.
- Use polymorphism.
- Understand and implement class hierarchies.
- Override methods.

Date: Monday, December 18, 2017

Please refer to the course outline addendum on the Learning Management System for further information.