# COURSE OUTLINE: CSD221 - JAVA II

Prepared: Fred Carella
Approved: Corey Meunier, Chair, Technology and Skilled Trades

| | |
|---|---|
| **Course Code: Title** | CSD221: JAVA II |
| **Program Number: Name** | 2090: COMPUTER PROGRAMMER |
| **Department:** | COMPUTER STUDIES |
| **Semesters/Terms:** | 21W |
| **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 |
| **Corequisites:** | There are no co-requisites for this course. |
| **This course is a pre-requisite for:** | CAR300, CSD309, CSD318, CSD322 |
| **Vocational Learning Outcomes (VLO's) addressed in this course:**<br><br>**Please refer to program web page for a complete listing of program outcomes where applicable.** | **2090 - COMPUTER PROGRAMMER**<br>VLO 1 Identify, analyze, develop, implement, verify and document the requirements for a computing environment.<br>VLO 2 Contribute to the diagnostics, troubleshooting, documenting and monitoring of technical problems using appropriate methodologies and tools.<br>VLO 4 Implement robust computing system solutions through validation testing that aligns with industry best practices.<br>VLO 5 Communicate and collaborate with team members and stakeholders to ensure effective working relationships.<br>VLO 6 Select and apply strategies for personal and professional development to enhance work performance.<br>VLO 8 Adhere to ethical, legal, and regulatory requirements and/or principles in the development and management of computing solutions and systems.<br>VLO 9 Support the analysis and definition of software system specifications based on functional and non-functional requirements.<br>VLO 10 Cntribute to the development, documentation, implementation, maintenance and testing of software systems by using industry standard software development methodologies based on defined specifications and existing technologies/frameworks.<br>VLO 11 Apply one or more programming paradigms such as, object-oriented, structured or functional programming, and design principles, as well as documented requirements, |

In response to public health requirements pertaining to the COVID19 pandemic, course delivery and assessment traditionally delivered in-class, may occur remotely either in whole or in part in the 2020-2021 academic year.

| | |
|---|---|
| | to the software development process. |
| **Essential Employability Skills (EES) addressed in this course:** | 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.<br>EES 2 Respond to written, spoken, or visual messages in a manner that ensures effective communication.<br>EES 4 Apply a systematic approach to solve problems.<br>EES 5 Use a variety of thinking skills to anticipate and solve problems.<br>EES 6 Locate, select, organize, and document information using appropriate technology and information systems.<br>EES 9 Interact with others in groups or teams that contribute to effective working relationships and the achievement of goals.<br>EES 10 Manage the use of time and other resources to complete projects.<br>EES 11 Take responsibility for ones own actions, decisions, and consequences. |
| **Course Evaluation:** | Passing Grade: 50%, D<br><br>A minimum program GPA of 2.0 or higher where program specific standards exist is required for graduation. |
| **Other Course Evaluation & Assessment Requirements:** | The student must pass both the lab and test portions of the course.<br><br>Attendance:<br>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.<br><br>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:<br><br>Course Hours Deduction<br>5 hrs/week (75 hrs) 1% / hr<br>4 hrs/week (60 hrs) 1.5% /hr<br>3 hrs/week (45 hrs) 2% /hr<br>2 hrs/week (30 hrs) 3%/hr<br><br>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.<br><br>Grade<br>Definition Grade Point Equivalent<br>A+ 90 - 100% 4.00<br>A 80 - 89%<br>B 70 - 79% 3.00 |

In response to public health requirements pertaining to the COVID19 pandemic, course delivery and assessment traditionally delivered in-class, may occur remotely either in whole or in part in the 2020-2021 academic year.

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.

**Books and Required Resources:**

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

Object Oriented Programming using Java by Simon Kendal
Publisher: Bookboon
https://bookboon.com/en/object-oriented-programming-using-java-ebook

**Course Outcomes and Learning Objectives:**

| **Course Outcome 1** | **Learning Objectives for Course Outcome 1** |
|---|---|
| 1. Define, describe and implement the various foundational elements of an object oriented system. | 1.1 Define and describe classes.<br>1.2 Define and describe objects .<br>1.3 Define and describe the various components of a class including attributes, accessors and mutators.<br>1.4 Define the relationship between classes and objects.<br>1.5 Create objects based on classes.<br>1.6 Define variables of various data types.<br>1.7 Define and implement programs that demonstrate variable scopes including static, local and class scope.<br>1.8 Define and call methods, with and without parameters.<br>1.9 Write and debug programs that demonstrate all of the above. |
| **Course Outcome 2** | **Learning Objectives for Course Outcome 2** |
| 2. Demonstrate an understanding of the components of an object oriented program. | 2.1 Write programs comprised of various objects and have those objects interact.<br>2.2 Demonstrate knowledge of and implement data using the various collection classes including lists and sets.<br>2.3 Demonstrate knowledge of and implement programs using various libraries.<br>2.4 Read class documentation.<br>2.5 Write class documentation.<br>2.6 Demonstrate an understanding of the package system and the structure of a project.<br>2.7 Test and debug programs using various methodologies such as unit testing and regression testing. |
| **Course Outcome 3** | **Learning Objectives for Course Outcome 3** |
| 3. Demonstrate an understanding of and | 3.1 Define and be able to demonstrate<br>3.1.1 Abstraction |

| | | |
|---|---|---|
| | implement the concepts of class design | 3.1.2 Encapsulation<br>3.1.3 Generalization<br>3.1.4 Polymorphism<br>3.1.5 Responsibility driven design.<br>3.1.6 Coupling<br>3.1.7 Cohesion<br>3.1.8 Refactoring |
| | **Course Outcome 4** | **Learning Objectives for Course Outcome 4** |
| | 4. Define and write programs that demonstrate inheritance and polymorphism. | 4.1 Define the benefits of inheritance and polymorphism and write programs that demonstrate each.<br>4.2 Use inheritance.<br>4.3 Use polymorphism.<br>4.4 Understand and implement class hierarchies.<br>4.5 Override methods. |

**Evaluation Process and Grading System:**

| Evaluation Type | Evaluation Weight |
|---|---|
| Labs | 40% |
| Tests | 60% |

**Date:** July 22, 2020

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