NORTHWESTERN POLYTECHNIC

**SCIENCE DEPARTMENT**

**COURSE OUTLINE – Spring 2025**

**CS 1150 (A3): Elementary Data Structures 3 (3-0-3) 06 Hours for 15 Weeks**

Northwestern Polytechnic acknowledges that our campuses are located on Treaty 8 territory, the ancestral and present-day home to many diverse First Nations, Metis, and Inuit people. We are grateful to work, live and learn on the traditional territory of Duncan's First Nation, Horse Lake First Nation and Sturgeon Lake Cree Nation, who are the original caretakers of this land.

We acknowledge the history of this land and we are thankful for the opportunity to walk together in friendship, where we will encourage and promote positive change for present and future generations.


| | | | |
|---|---|---|---|
| **INSTRUCTOR:** | Ubaid Abbasi | **PHONE:** | 780-539-2016 |
| **OFFICE:** | C-427 | **E-MAIL:** | UAbbasi@nwpolytech.ca |
| **OFFICE HOURS:** | 11:30-12:30 Monday or appointment by email | | |


**CALENDAR DESCRIPTION:**

The course provides a review of programming principles (specification, implementation and testing), and an extension of object-oriented concepts from CS1140 including data abstraction, modular program construction and program reuse. The emphasis is on dynamic data structures (e.g. lists, string, stacks, queues), and their associated algorithms (e.g. recursion, traversal, sorting, searching, hashing).

**PREREQUISITE(S)/COREQUISITE:** CS1140

**REQUIRED TEXT/RESOURCE MATERIALS:**

Introduction to Java Programming 12th Edition, Comprehensive Version, By Y. Daniel Liang, Pearson Publishing, ISBN 12th Edition 9780136520238

The 11th edition is acceptable
ISBN 11th Edition 9780134671048

**Note:** Additional handouts will be provided in class.

**DELIVERY MODE(S):**

This course includes 3-hours of lecture per week and a 3-hour lab per week

| | | | |
|---|---|---|---|
| **Lectures:** | G112 | Tuesday, Thursday | 09:00 - 11:50AM |
| **Labs:** | G112 | Tuesday, Thursday | 12:00 - 02:50PM |

**LEARNING OUTCOMES:**

By taking this course, students will gain the ability to:

- Analyze problems, design algorithms and data structures to implement computational solutions to problems using an object-oriented computer language.
- Design and implement object-oriented classes, using inheritance and polymorphism.
- Design and implement array based and linked data structures like strings, stacks, queues, lists, trees, heaps, sets, dictionaries and graphs.
- Describe and implement common algorithms related to searching, sorting, traversals, and hashing.

**TRANSFERABILITY:**

**UA, UC, UL, AU, KUC, GMU.**

***Warning:** Although we strive to make the transferability information in this document up-to-date and accurate, **the student has the final responsibility for ensuring the transferability of this course to Alberta Colleges and Universities**. Please consult the Alberta Transfer Guide for more information. You may check to ensure the transferability of this course at Alberta Transfer Guide main page http://www.transferalberta.ca or, if you do not want to navigate through few links, at http://alis.alberta.ca/ps/tsp/ta/tbi/onlinesearch.html?SearchMode=S&step=2

** Grade of D or D+ may not be acceptable for transfer to other post-secondary institutions. **Students are cautioned that it is their responsibility to contact the receiving institutions to ensure transferability**

**EVALUATIONS:**

Your final grade will be determined in the following manner:

| | |
|---|---|
| **Lab Assignments** | **20%** |
| **Quizzes(2-4)** | **20%** |
| **Midterm** | **25%** |

**Final Exam** **35%**

**GRADING CRITERIA: (The following criteria may be changed to suite the particular course/instructor)**

Please note that most universities will not accept your course for transfer credit **IF** your grade is **less than C-**.

| Alpha Grade | 4-point Equivalent | Percentage Guidelines | Alpha Grade | 4-point Equivalent | Percentage Guidelines |
|---|---|---|---|---|---|
| A+ | 4.0 | 95-100 | C+ | 2.3 | 67-69 |
| A | 4.0 | 85-94 | C | 2.0 | 63-66 |
| A- | 3.7 | 80-84 | C- | 1.7 | 60-62 |
| B+ | 3.3 | 77-79 | D+ | 1.3 | 55-59 |
| B | 3.0 | 73-76 | D | 1.0 | 50-54 |
| B- | 2.7 | 70-72 | F | 0.0 | 00-49 |

**COURSE SCHEDULE/TENTATIVE TIMELINE:**

| Sequence | Topic |
|---|---|
| Week 1 | **Objects and Classes: Chapter 9**<br>• Defining Classes & Creating Objects<br>• Constructors, Static Variables and Methods<br>• Visibility Modifiers, Data Fields Encapsulation<br>• Array of Objects and Scope of Variables |
| Week 2 | **Object Oriented Thinking: Chapter 10**<br>• Class Abstraction & Encapsulation<br>• Objects and Class Relationships<br>• Primitive Types and Wrapper Class Types<br>• String Class |
| Week 3,4 | **Quiz 1**<br>**Inheritance and Polymorphism: Chapter 11**<br>• Superclasses and Subclasses<br>• Overriding and Overloading |

| | |
|---|---|
| | • Polymorphism<br>• Dynamic Binding<br>• Protected Data and Methods<br>• Preventing Extending and Overriding |
| Week 5 | **Exception Handling : Chapter 12**<br>• Exception Types<br>• Use of Exceptions<br>• Re-throwing Exceptions and Chained Exceptions<br>• Custom Exception Classes |
| Week 6 | **Abstract Classes and Interfaces: Chapter 13**<br>• Abstract Classes<br>• Interfaces<br>• Class Design Guidelines |
| Week 7 | **Generics: Chapter 19**<br>• Defining Generic Classes and Interfaces<br>• Generic Methods<br>• Raw Types and Backward Compatibility<br>• Wildcard Generic Types<br>• Restriction in Generics |
| Week 8 | **Review + Midterm** |
| Week 9 | **Developing Efficient Algorithms: Chapter 22**<br>• Algorithm Efficiency and Big O Notation<br>• Analyzing Algorithm Time Complexity<br>• Determining Big O<br>• Introduction to Dynamic Programming |
| Week 10,11 | **Linked Lists, Stack and Queues: Chapter 24**<br>• Common Operations for Lists<br>• Array Lists<br>• Linked Lists<br>• Stack and Queues<br>• Priority Queues<br>**Quiz 2** |
| Week 12,13 | **Recursion, Searching and Sorting: Chapter 18, 23, 25**<br>• Recursion<br>• Insertion Sort, Bubble sort, Merge Sort, Quick Sort and Heap Sort<br>• Binary Search Trees |

| | |
|---|---|
| Week 14,15 | **Introduction to Hashing and Graphs: Chapter 27 and Chapter 28** |

**STUDENT RESPONSIBILITIES:**

- The Student must pass the theory/concepts portion of the course in order to qualify for a passing grade for the term. In other words, a student must obtain 40 out of a possible 80 points (from exams/quizzes) before adding the lab assignment marks to compute the final grade. **If you cannot achieve the required 50% (on exams) then regardless of your lab assignment grades, you cannot pass the course.**
- No late assignments will be accepted. The student is responsible for adhering to all requirements as specified for each assignment.
- When necessary, lab time may be utilized for lecturing on specific Java features. The remainder of the lab time will generally be used as "hands-on" programming time.

**STATEMENT ON PLAGIARISM AND CHEATING:**

Academic Misconduct will not be tolerated. For a more precise definition of academic misconduct and its consequences, refer to the Student Rights and Responsibilities policy available at https://www.nwpolytech.ca/about/administration/policies/index.html.

**Note: all Academic and Administrative policies are available on the same page.