



## DELIVERY MODE(S):

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

<b>Lectures:</b>	J204	Monday, Wednesday	10:00 - 11:20AM
<b>Labs:</b>	E306	Monday	02:30 – 05:30PM
	E306	Tuesday	02:30 – 05:30PM

## 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.transferralberta.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:

<b>Lab Assignments</b>	<b>20%</b>
<b>Quizzes(2-4)</b>	<b>20%</b>
<b>Midterm</b>	<b>25%</b>
<b>Final Exam</b>	<b>35%</b>

**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	<b>Objects and Classes: Chapter 9</b> <ul style="list-style-type: none"> <li>Defining Classes &amp; Creating Objects</li> <li>Constructors, Static Variables and Methods</li> <li>Visibility Modifiers, Data Fields Encapsulation</li> <li>Array of Objects and Scope of Variables</li> </ul>
Week 2	<b>Object Oriented Thinking: Chapter 10</b> <ul style="list-style-type: none"> <li>Class Abstraction &amp; Encapsulation</li> <li>Objects and Class Relationships</li> <li>Primitive Types and Wrapper Class Types</li> <li>String Class</li> </ul>
Week 3,4	<p style="text-align: center;"><b>Quiz 1</b></p> <b>Inheritance and Polymorphism: Chapter 11</b> <ul style="list-style-type: none"> <li>Superclasses and Subclasses</li> <li>Overriding and Overloading</li> <li>Polymorphism</li> </ul>

	<ul style="list-style-type: none"> <li>• Dynamic Binding</li> <li>• Protected Data and Methods</li> <li>• Preventing Extending and Overriding</li> </ul>
Week 5	<b>Exception Handling : Chapter 12</b> <ul style="list-style-type: none"> <li>• Exception Types</li> <li>• Use of Exceptions</li> <li>• Re-throwing Exceptions and Chained Exceptions</li> <li>• Custom Exception Classes</li> </ul>
Week 6	<b>Abstract Classes and Interfaces: Chapter 13</b> <ul style="list-style-type: none"> <li>• Abstract Classes</li> <li>• Interfaces</li> <li>• Class Design Guidelines</li> </ul>
Week 7	<b>Generics: Chapter 19</b> <ul style="list-style-type: none"> <li>• Defining Generic Classes and Interfaces</li> <li>• Generic Methods</li> <li>• Raw Types and Backward Compatibility</li> <li>• Wildcard Generic Types</li> <li>• Restriction in Generics</li> </ul>
Week 8	<b>Review + Midterm</b>
Week 9	<b>Developing Efficient Algorithms: Chapter 22</b> <ul style="list-style-type: none"> <li>• Algorithm Efficiency and Big O Notation</li> <li>• Analyzing Algorithm Time Complexity</li> <li>• Determining Big O</li> <li>• Introduction to Dynamic Programming</li> </ul>
Week 10,11	<b>Linked Lists, Stack and Queues: Chapter 24</b> <ul style="list-style-type: none"> <li>• Common Operations for Lists</li> <li>• Array Lists</li> <li>• Linked Lists</li> <li>• Stack and Queues</li> <li>• Priority Queues</li> </ul> <p style="text-align: center;"><b>Quiz 2</b></p>
Week 12,13	<b>Recursion, Searching and Sorting: Chapter 18, 23, 25</b> <ul style="list-style-type: none"> <li>• Recursion</li> <li>• Insertion Sort, Bubble sort, Merge Sort, Quick Sort and Heap Sort</li> <li>• Binary Search Trees</li> </ul>
Week 14,15	<b>Introduction to Hashing and Graphs: Chapter 27 and Chapter 28</b>

## **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.