

Summary of Alberta Education CSE Courses & INF 1030

Alberta Education Introductory Level (Grade 10) CSE Courses			
Num	Course Name	Prereq	CSE Specific Outcomes
1010	COMPUTER SCIENCE 1	None	<ol style="list-style-type: none"> 1. Describe CS nature & approaches and areas of interest. 2. Basic algorithms sequential input, process & output. 3. Explain & demo structured programming 4. Explain evolution, types and role of programming languages. 5. Explain a von Neumann computer system.
1110	STRUCTURED PROGRAM 1	None	<ol style="list-style-type: none"> 1. Write sequential algorithms to solve problems with input, process & output. 2. Write source code, convert to ML, execute and debug. 3. Compare results with intent of the algorithm and modify as required.
1120	STRUCTURED PROGRAM 2	1110	<ol style="list-style-type: none"> 1. Write & execute algorithms & programs involving selection & iteration.
1210	SCRIPTING 1	None	<ol style="list-style-type: none"> 1. Markup (HTML, etc) code.
1220	SCRIPTING 2	None	<ol style="list-style-type: none"> 1. Compare static and dynamic client-side scripts.
1240	ROBOTICS PRG 1	1110	
1910	CSE PROJECT A	Connects with 2+ CSE courses, one of which must be at the introductory level.	

Alberta Education Introductory Level (Grade 10) INF Courses			
Num	Course Name	Prereq	CSE Specific Outcomes
INF 1030	WORD PROCESSING 1	None	<ol style="list-style-type: none"> 1. Create & customize, format, organize and review documents & visual content. 2. Apply consistent & appropriate work station routines.
1910	INF PROJECT A	Connects with 2+ courses, one at INF introductory level: (INF 1030) & another (CSE ??).	

Alberta Education Intermediate Level (Grade 11) CSE Courses			
Num	Course Name	Prereq	CSE Specific Outcomes
2010	COMPUTER SCIENCE 2	1010 1120	<ol style="list-style-type: none"> 1. Past, Present & Future Developments in Computer Technology. 2. Explain & demo key approaches of modular programming paradigm. 3. Procedural abstraction & Local Scope == Methods & Objects. 4. Demonstrate traversing, searching, sorting & merging. 5. Demonstrate arrays, vectors, matrices, enumerated data, records (objects). 6. Demonstrate text files. 7. Demonstrate CPU, ALU, registers, bus, memory, opcodes, operands, symbols: Do the Machine Instruction Cycle: Fetch-Execute-Decode-Store Cycle.
2110	PROCEDURAL PROGRAM 1	1120	<ol style="list-style-type: none"> 1. Modular programming Translate algorithms into methods, execute, debug.
2120	DATA STRUCTURES 1	2110	<ol style="list-style-type: none"> 1. Fundamental data types (also primitives) 2. Array (single, double, parallel) – create, insert, search, size, copy, compare.
2130	FILES & FILE STRUCTURES 1	2120	<ol style="list-style-type: none"> 1. Store & retrieve data on disks operations 2. Sequential, random, indexed (ISAM) file handling
2140	SECOND LANG PROGRAM 1	1120 or 2110	<ol style="list-style-type: none"> 1. Write IPO & “modular” structured algorithms: methods (1 & 2 way)
2210	SCRIPTING 3	1120 1220	<ol style="list-style-type: none"> 1. Scriptlets, etc.
2240	ROBOTICS PRG 2	1120 1240	<ol style="list-style-type: none"> 1. procedural programming techniques and fundamental data structures
2910	CSE PROJECT B	Connects with 2+ CSE courses, one of which must be at the intermediate level.	
2920	CSE PROJECT C	Connects with 2+ CSE courses, one of which must be at the intermediate level.	
2950	CSE INTERMEDIATE PRACTICUM	None	<ol style="list-style-type: none"> 1. Accessed only by students continuing to work toward attaining a recognized credential/credentials or an articulation offered by an external organization. 2. Connects with 1+ CSE courses in the CSE occupational area.

Alberta Education Advanced Level (Grade 12) CSE Courses

Num	Course Name	Prereq	CSE Specific Outcomes
3010	COMPUTER SCIENCE 3	2010 2110	<ol style="list-style-type: none"> 1. History & nature of information revolution & knowledge-based society. 2. Explain nature, rationale and key approaches associated with OOP. 3. Demonstrate object-oriented design techniques. 4. Binary & hex number systems, data encoding, logic gates & digital computer.
3020	COMPUTER SCIENCE 4	3010 3110	<ol style="list-style-type: none"> 1. Structure, utility, operations of dynamic abstract data types (ADTs). 2. Asymptotic analyses of algorithmic efficiency == Big O. 3. Rationale and use of recursive and introductory recursive operations. 4. Nature, architecture, operation and utility of a Turing machine. 5. Presentation re computer science's impact on society.
3110	ITERATIVE ALGORITHM 1	2120	<ol style="list-style-type: none"> 1. Binary search & Bubble, Insertion, Selection sorts & a simple merge.
3120	OOP 1	2110	<ol style="list-style-type: none"> 1. Features of OOP: encapsulation, modularity, polymorphism, inheritance.
3130	OOP 2	3120	<ol style="list-style-type: none"> 1. UML Design.
3140	SECOND LANG PROGRAMMING 2	2120 (Data Struct)	<ol style="list-style-type: none"> 1. Fundamental Data Structures.
3210	SERVER-SIDE SCRIPTING 1	2210 2110 2120	<ol style="list-style-type: none"> 1. Dynamic Web sites. 2. Write & debug scripts.
3240	ROBOTICS PRG 3	2240 2110	<ol style="list-style-type: none"> 1. Apply OOP to robotics programming. 2. Create class libraries in robotics programming. 3. Use OOP in robot, control systems & environment set of predetermined tasks.
3310	RECURSIVE ALGORITHMS 1	3110 3120	<ol style="list-style-type: none"> 1. Recursive functions & procedures. 2. Recursive binary search & heap sort & merge sort.
3320	DYNAMIC DATA STRUCTURES 1	3310	<ol style="list-style-type: none"> 1. Linked Lists: create, insert node, delete, replace, traverse, size, & find data.
3330	DYNAMIC DATA STRUCTURES 2	3320	<ol style="list-style-type: none"> 1. Stacks & Queues. 2. Unordered data types: set and map. 3. Create & copy data structure. 4. Push, pop and peek (Stacks) & Enqueue & Dequeue (Queues). 5. Link keys and values (Maps). 6. Search, insert, remove and modify data elements (Sets &/or Maps). 7. Determine equity between sets. 8. Determine union, intersection, difference & Symmetric Difference. 9. Delete data structure.
3340	DYNAMIC DATA STRUCTURES 3	3330	<ol style="list-style-type: none"> 1. Trees: create, copy, delete. 2. Traversals: preorder, in-order, post-order. 3. Search, insert, remove & modify 4. List data elements accumulated by a tree transversal. 5. List the pop & heapify operation for heaps.
3910	CSE PROJECT D	Connects with 2+ CSE courses, one of which must be at the advanced level.	
3920	CSE PROJECT E	Connects with 2+ CSE courses, one of which must be at the advanced level.	
2950	CSE ADVANCED PRACTICUM	None	<ol style="list-style-type: none"> 1. Accessed only by students continuing to work toward attaining a recognized credential/credentials or an articulation offered by an external organization. 2. Connects with 1+ CSE courses in the CSE occupational area.