CST8110 – Introduction to Programming
Course Section Information F09

Course Professor: Linda Crane

Lab Professors: Robert Brandon, John Synowski, Stanley Pieda

Week 1	***Overview of Course and Policies*** ▪ Course Outline. ▪ Plagiarism, Cheating etc. ▪ Use of Cell Phones and pagers in class and labs. ▪ Use of Palm type devices. ▪ College, School and Departmental Policies. ▪ Coordinators and their roles.	Hybrid Activity #0 - due Sept 22nd in lecture class
Week 1	***Overview of Computers – Chapter 1.3, 1.6*** ▪ Hardware components ▪ Software components – High level languages vs Assembler	Lab #1 due in lab next week
Week 1	***Java Environment – Chapter 2, 3*** ▪ Compiler/run-time environment ▪ Eclipse as tool ▪ Algonquin environment	Lab #1 due in lab next week
Week 2, 3	***Problem Solving – Chapter 4.2, 4.3*** ▪ Define the problem in a logical manner. ▪ Ask the right questions. ▪ Create a narrative or prose definition of the problem captured in the *Problem Statement* ***Problem definition to Algorithm - Chapter 4.2, 4.3*** ▪ What is an algorithm? ▪ Narrative or prose solution. ▪ Step wise refinement of solution. ▪ Write the algorithm. ▪ Use PDL (Program Description Language) to write the solution in a structured fashion. ▪ Procedure-based solutions. ***Desk Check*** ▪ Develop a desk-check for the solution. ▪ Walk through the algorithm using the desk-check plan. ***Documentation and Testing*** ▪ Computer Studies Department submission standard.	Hybrid Activity #1 due – Sept 22nd in lecture class Lab #1 and #2 due in lab

	Documents required before the coding begins.	
Week 3, 4	***Data representation in a computer system – Chapter 2*** ▪ *int* (integer), *char* (character) and *float, double* (floating-point data types). ▪ Characteristics of the data types (size, format, range and precision). ▪ Declare data type variables in Java ▪ Representation of variables in memory. ▪ Arithmetic overflow and underflow for integer variables.	Hybrid Activity #2 due Sept 29th in lecture class Lab #3 due in lab **Assignment #1 due – Monday Sept 28th at 11pm.**
Week 4, 5	***Instruction format in Java – Chapter 2*** ▪ Assignment Statements. ▪ Console input / output operations in Java ▪ Role of the Compiler and RTE. ***IDE and Debugger*** Note: The debugger is introduced at the earliest possible stage to let students have the concrete experience of viewing the representation of variables, and the sequence of execution. Advanced components of the debugger will be introduced on an as-needed basis throughout the course. ▪ What is the debugger? ▪ How do you use the debugger? ▪ Breakpoints. ▪ Variable Monitor: *Local*, *Auto*, *Watch*. ▪ Modifying memory at run-time. ▪ *Step into*. ▪ *Step over*. ▪ *Call stack*.	Hybrid Activity #3 due Oct 6th in lecture class Lab #4 due in lab
Week 6	***Selection – Chapter 4*** ▪ *if* construct. ▪ *if – else* construct. ▪ *if – else if* construct. ▪ *switch* – case construct.	Hybrid Activity #4 due Oct 13th in lecture class
Week 7	***Iteration- Chapter 5*** ▪ *while* construct. ▪ *do – while* construct. ▪ *for* construct. ▪ *break* statement.	Hybrid Activity #5 due Oct 20th in lecture class Lab #5 due in lab **Assignment #2 due – Monday October 19th at 11pm.**
Week 8	***Test Plan for the Solution: User Perspective*** ▪ Identify normal user input. (What are the	**Midterm #1 in class October 27th**

	reasonable expected input values?) ▪ Identify out-of-range user input. (What are unreasonable input values, but ones that an uniformed or mischievous user might input?) ***Test Plan for the Solution: Programmer Perspective*** ▪ Determine boundaries for variables (based on the range of possible user input). ▪ Develop the test plan data. ▪ Use the debugger to validate the expected results (as defined in the test plan). ▪ Monitor data changes. ▪ Document the test plan.	
Week 9, 10	***Java Library Methods – Chapter 3,6*** ▪ What is a Library method? ▪ How do you obtain details of a Library method? ▪ What do the manual pages tell us? ▪ What are *arguments* and the *return* value? ***Methods – Chapter 3, 6*** ▪ Methods with no parameters or return. ▪ Write a method in Java ▪ Methods with parameters and no return. ▪ Methods with parameters and a return. ▪ Note: At this point students will be use the debugger Call Stack viewer to have the concrete experience of function execution.	Hybrid Activity #6 due Nov 3rd in lecture class Lab #6 due in lab **Assignment #3 due – Monday Nov 2nd at 11pm.** Hybrid Activity #7 due Nov 10th in lecture class Lab #7 due in lab

Week10,11, 12	***Classes – Chapter 3,6***
	▪ Use existing *string* class.
	  ▪ Create *string* objects.
	  ▪ Use *string* objects to call *string* class member functions.
	▪ Create programmer-defined class.
	  ▪ Combine data and methods.
	  ▪ Data members are usually *private*.
	  ▪ Member functions are usually *public*.
	  ▪ Use Unified Modeling Language's (UML) class diagram to express the static structure of classes.
	  ▪ Implement *Accessor* / *Mutator* (i.e. Get / Set) design patterns.
	▪ Create *class* objects.
	▪ Use *class* objects to access methods.
		Hybrid Activity #8 due Nov 17th in lecture class
		Assignment #4 due – Monday Nov 16th at 11pm.

Week12,13, 14	***Arrays – Chapter 7*** ▪ Define one-dimensional arrays of *char* for simple character strings. ▪ Use one dimensional arrays of *char*. ▪ Define one-dimensional arrays of *int*. ▪ Use one dimensional arrays of *int*. ▪ Define one-dimensional arrays of programmer-defined class. ▪ Use one dimensional arrays of programmer-defined class.	**Midterm #2 in class November 24th** Lab #8 due in lab Hybrid Activity #9 due Dec 1st in lecture class Lab #9 due in lab **Assignment #5 due – Monday Dec 7th at 11pm.**
Week 15	FINAL EXAM week	