Douglas College

Course Information

Page 1 of 4

Course Number Descriptive Title Credits	A: Division Instructional	Date June 18, 1997
C: CMPT-110 D: Introduction to Computing Science Using C++ Descriptive Title F: Calendar Description This course introduces the science of computing. Emphasis is placed on the analysis of problems, the design of algorithms, and the abstraction of control and data in computer implementations of the design. Initially structured top-down design and procedural programming is used followed by an introduction to recursive functional programming then an introduction to object Oriented Design (OOD) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture Lab. 1 x 2 hrs/week	B: Department Pure and Applied Science and Technology	
C: CMPT-110 D: Introduction to Computing Science Using C++ E: 4 Course Number F: Calendar Description Figure Title F: Calendar Description F: Calendar Description Figure Title F: Calendar Description Feet Submary of Revisions Feeting Title Feet Submary of Revisions Feeting Title Feet Submary of Revisions Feeting Title Feet A. K.		Revision of Course X
Course Number Descriptive Title Credits F: Calendar Description This course introduces the science of computing. Emphasis is placed on the analysis of problems, the design of algorithms, and the abstraction of control and data in computer implementations of the design. Initially structured top-down design and procedural programming is used followed by an introduction to recursive functional programming then an introduction to Object Oriented Design (OOD) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture Lab. Lab. Lx 2 hrs/week Lab. Lx 2 hrs/biweekly Seminar Clinical Experience Practicum Shop Studio Student Directed Learning Student Directed Learning Total L: College Credit Transfer Transfer Total L: College Credit Transfer Registrar's Office - N.W. By: JUL - 3 1997 Registrar's Office - N.W. By: Others Course Gauivalents U.B.C. CPSC(3) S.F.U. CMPT-101 Lof Vic. CSC110 Others intransfer guide		Dated Jan 7, 1991
Course Number Descriptive Title Credits F: Calendar Description This course introduces the science of computing. Emphasis is placed on the analysis of problems, the design of algorithms, and the abstraction of control and data in computer implementations of the design. Initially structured top-down design and procedural programming is used followed by an introduction to recursive functional programming then an introduction to Object Oriented Design (OOD) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture Lab. Lix 2 hrs/week Lab. Lix 2 hrs/biweekly Seminar Clinical Experience Practicum Shop Studio Student Directed Learning Student Directed Learning Total L: College Credit Transfer Transfer Total L: College Credit Transfer Registrar's Office - N.W. By: JUL - 3 1997 Registrar's Office - N.W. By: Others Course Gauivalents U.B.C. CPSC(3) S.F.U. CMPT-101 Lu fyic. CSC110 Others intransfer guide		
F: Calendar Description This course introduces the science of computing. Emphasis is placed on the analysis of problems, the design of algorithms, and the abstraction of control and data in computer implementations of the design. Initially structured top-down design and procedural programming is used followed by an introduction to Object Oriented Design (OOD) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture	C: CMPT-110 D: Introduction to Computing Science	ence Using C++ E: 4
This course introduces the science of computing. Emphasis is placed on the analysis of problems, the design of algorithms, and the abstraction of control and data in computer implementations of the design. Initially structured top-down design and procedural programming is used followed by an introduction to recursive functional programming then an introduction to Object Oriented Design (OOP) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture	Course Number Descriptive Title	Credits
This course introduces the science of computing. Emphasis is placed on the analysis of problems, the design of algorithms, and the abstraction of control and data in computer implementations of the design. Initially structured top-down design and procedural programming is used followed by an introduction to recursive functional programming then an introduction to Object Oriented Design (OOP) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture		
placed on the analysis of problems, the design of algorithms, and the abstraction of control and data in computer implementations of the design. Initially structured top-down design and procedural programming is used followed by an introduction to recursive functional programming then an introduction to Object Oriented Design (OOD) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture 2 x 2 hrs/week	F: Calendar Description	Summary of Revisions
abstraction of control and data in computer implementations of the design. Initially structured top-down design and procedural programming is used followed by an introduction to recursive functional programming then an introduction to Object Oriented Design (OOD) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture	This course introduces the science of computing. Emphasis is	
design. Initially structured top-down design and procedural programming is used followed by an introduction to recursive functional programming then an introduction to Object Oriented Design (OOD) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture		
programming is used followed by an introduction to recursive functional programming then an introduction to Object Oriented Design (OOD) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture		f the K
functional programming then an introduction to Object Oriented Design (OOD) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture Lab. Lab. Lx 2 hrs/biweekly Seminar Clinical Experience Practicum Shop Studio Student Directed Learning Other Total L: Course Frerequisites: CMPT-100 and MATH-110 I: Course Corequisites None Practicum Shop Studio Student Directed Learning Other Total L: College Credit Transfer X DECEIVE M: Transfer Credit Requested Granted X Course Equivalents U.B.C. CPSC(3) S.F.U. CMPT101 U. of Vic. CSC110 Others in transfer guide	design. Initially structured top-down design and procedural	
Design (OOD) and Object Oriented Programming (OOP). C++ is used as the implementation language. G: Type of Instruction Lecture		
used as the implementation language. G: Type of Instruction Lecture Lab. Lab. Seminar Clinical Experience Practicum Shop Studio Student Directed Learning Other Total L: College Credit Transfer _X JUL - 3 1997 Registrar's Office - N.W. By:	functional programming then an introduction to Object Orient	ed
G: Type of Instruction Lecture		+ is
Lecture Lab. Seminar Clinical Experience Practicum Shop Studio Student Directed Learning Other Total L: College Credit TransferX JUL - 3 1997 Registrar's Office - N.W. Ry: CMPT-100 and MATH-110 I: Course Corequisites None J: Course for which this course is a prerequisite CMPT-210, CMPT-220 K: Maximum Class Size 34 Course Equivalents U.B.C. CPSC(3) S.F.U. CMPT101 U. of Vic. CSC110 Others in transfer guide		
Lecture Lab. Seminar Clinical Experience Practicum Shop Studio Student Directed Learning Other Total L: College Credit Transfer _ X College Credit Col	G: Type of Instruction	
Lab.	Testure 2 21 / 1	
Seminar Clinical Experience Practicum Shop Studio Student Directed Learning Other Total L: College Credit Transfer X JUL - 3 1997 Registrar's Office - N.W. Ry: I: Course Corequisites None I: Course for which this course is a prerequisite CMPT-210, CMPT-220 K: Maximum Class Size 34 M: Transfer Credit Requested Granted X Course Equivalents U.B.C. CPSC(3) S.F.U. CMPT101 L. of Vic. CSC110 Others in transfer guide		MATH-110
Clinical Experience Practicum Shop Studio Student Directed Learning Other Total L: College Credit Transfer X Transfer X Total JUL - 3 1997 Registrar's Office - N.W. Ry: Shop Student Directed Learning Shrs/week (approx.) JUL - 3 1997 Registrar's Office - N.W. Ry: None J: Course for which this course is a prerequisite CMPT-210, CMPT-220 K: Maximum Class Size 34 M: Transfer Credit Requested Granted X Course Equivalents U.B.C. CPSC(3) S.F.U. CMPT101 II. of Vic. CSC110 Others in transfer guide		
Practicum Shop Studio Studio Student Directed Learning Other Total L: College Credit Transfer X JUL - 3 1997 Registrar's Office - N.W. By: Typical Student Directed Learning Shrs/week (approx.) J: Course for which this course is a prerequisite CMPT-210, CMPT-220 K: Maximum Class Size 34 M: Transfer Credit Requested Granted X Course Equivalents U.B.C. CPSC(3) S.F.U. CMPT101 UL of Vic. CSC110 Others in transfer guide		
Shop Studio Student Directed Learning Other Total L: College Credit Transfer X JUL - 3 1997 Registrar's Office - N.W. Ry: JUL - 3 1997 Registrar's Office - N.W. Ry: JUL - 3 1997 JUL - 3 1997 Registrar's Office - N.W. JUL - 3 1997 JUL - 3 199		None
Studio Student Directed Learning Other Total L: College Credit Transfer X JUL - 3 1997 Registrar's Office - N.W. Ry: COURSE is a prerequisite CMPT-210, CMPT-220 K: Maximum Class Size 34 M: Transfer Credit Requested Granted X Course Equivalents U.B.C. CPSC(3) S.F.U. CMPT101 U. of Vic. CSC110 Others in transfer guide	Practicum	7. 6. 1.1.1.
Student Directed Learning Other Total L: College Credit Transfer _ X JUL - 3 1997 Registrar's Office - N.W. By:	Shop	
Other Total L: College Credit Transfer X JUL - 3 1997 Registrar's Office - N.W. By: Student Directed Learning 5 hrs/week (approx.) K: Maximum Class Size 34 M: Transfer Credit Requested Granted X Course Equivalents U.B.C. CPSC(3) S.F.U. CMPT101 U. of Vic. CSC110 Others in transfer guide	Studio	
Total 10 hrs/week L: College Credit Transfer X JUL - 3 1997 Registrar's Office - N.W. By:	Student Directed Learning _5 hrs/week (approx.)	CMP1-210, CMP1-220
Total L: College Credit Transfer X DECEIVE M: Transfer Credit Requested Granted X Course Equivalents U.B.C. CPSC(3) S.F.U. CMPT101 U. of Vic. CSC110 Others in transfer guide		V: Maximum Class Size
Total L: College Credit Transfer X DECEIVE M: Transfer Credit Requested Granted X Course Equivalents U.B.C. CPSC(3) S.F.U. CMPT101 U.of Vic. CSC110 Others in transfer guide		
L: College Credit Transfer _X	Total 10 hrs/week	34
Transfer X DECEIVE Requested		M: Transfor Credit
TUL - 3 1997 Registrar's Office - N.W. Ry:	Transfer X	· · · · · · · · · · · · · · · · · · ·
TUL - 3 1997 Registrar's Office - N.W. Ry:	UFCEIVEN	•
TUL - 3 1997 Registrar's Office - N.W. Ry:		Grance A
Registrar's Office - N.W. Ry:		Course Equivalents
Registrar's Office - N.W. By:	JUL - 3 1997	•
Registrar's Office - N.W. Ry: Others in transfer guide	D . 4.:	
Others in transfer guide	Registrar's Office - N.W.	I
Think the wal	By: ************************************	
Course Designer Vice-President - Instruction		
Course Designer Vice-President - Instruction		Attito 11/h las to
Course Designer // Vice-President - Instruction	JANN (MUTULES WILL
	Course Designer	Vice-President - Instruction
allamond Wolson IT. H. Why	allamond Wilson	T. H. WYS
Dean Registrar	Dean	Registrar

N: Textbook and Materials to be Purchased by Students

- Dale N., Weems C., Headington M., <u>Programming and Problem Solving with C++</u>,
 D. C. Heath and Company
- Portfolio for Programming Assignments
- Two 31/2" high density diskettes

O: Course Objectives

The student should be able to:

- analyze problem specifications
- form data and control abstractions
- design computer algorithms using either a structured top down design methodology or Object Oriented Design
- implement, in a widely acceptable style, algorithms in C++ using standard programming methodologies
- document a project

The student should understand the concepts of:

- generality through abstractions
- maintainability, reusability and extensibility through modularity

P: Course Content

- 1 Introduction and Review (syntax of C++)
 - 1.1 Program Structure
 - 1.2 Primitive data types and expressions
 - 1.3 Control Structures
 - 1.4 Functions and parameter passing
 - 1.5 Arrays
 - 1.6 Top-down design review and specs. for assignment #1: procedural programming with emphasis on control structures, procedures, and arrays
- 2 Abstractions (implementations are not considered)
 - 2.1 Strings
 - 2.2 Collections
 - 2.2.1 Lists
 - 2.2.2 Sets
 - 2.2.3 Stacks
- 3 Implementing Abstractions
 - 3.1 C++ Strings
 - 3.2 Introduction to pointers (domain of arrays and parameter passing)
 - 3.3 C++ records (struct)
 - 3.4 Structured design issues and specs. for assignment #2: procedural programming with emphasis on: cohesion and coupling and using more complicated static data structures
 - 3.5 Design of set primitives
 - 3.6 Recursion
 - 3.6.1 Numerical examples: factorial, Fibonacci, ...
 - 3.6.2 Examples from symbolic (LISP-like) expressions (SExpressions)
 - 3.7 Discussion and specs for assignment #3: functional programming using an existing module for SExpressions with emphasis on recursion and list processing
- 4 Encapsulation, Instantiation, and OOP
 - 4.1 Structure (syntax and semantics)
 - 4.2 Examples
 - 4.2.1 Sets implementation and use
 - 4.2.2 Stacks implementation and use
 - 4.3 Specs for assignment #4: OOP
- 5 OOD and Separate Compilations
 - 5.1 OOD
 - 5.2 Examples
 - 5.3 Specs for assignment #5: OOP
 - 5.4 Introduction to inheritance
 - 5.4.1 Examples

Q: Method of Instruction

There are three components to the course: lectures, labs., and assignments.

The lecture is used to introduce new material; usually via a sequence of theoretical concepts, practical considerations (usually language dependent), and one or more example case studies. The book is to be used as an additional source of problems and examples.

The two hour biweekly lab. is exclusively used to evaluate the student's practical programming ability. They are marked mostly on results; i.e. correctness of the algorithm.

Assignments are the most important learning vehicle and are done on the student's own time. They are marked according to program design, correctness and efficiency of the algorithms, coding style, and completeness of the documentation.

R: Evaluation

The final grade will be calculated from a particular distribution from the range below. The exact distribution will be given to the student on the first day of classes along with the course outline and necessary policies.

Distribution Range:

6 labs.	=	15% - 25%
2 tests @ 15% - 20% each	=	30% - 40%
1 exam	==	20% - 30%
5 assignments	=	20% - 35%
Example Distribution:		
6 labs.	= -	15%
test #1		15%
test #2	=	20%
assignments	=	25%
exam	=	25%
Total	=	100%

[©] Douglas College. All Rights Reserved.