

CSE 4252 (Approved): Programming in C++

Course Description

Syntax and pragmatics of C++ programming; C++ types, arrays, classes, pointers; objects and classes; compile-time vs. run-time picture; inheritance; template classes.

Prior Course Number: CSE 459.22

Transcript Abbreviation: C++ Prgrmng

Grading Plan: Satisfactory/Unsatisfactory

Course Deliveries: Classroom

Course Levels: Undergrad

Student Ranks: Junior, Senior

Course Offerings: Autumn

Flex Scheduled Course: Never

Course Frequency: Every Year

Course Length: 14 Week

Credits: 1.0

Repeatable: No

Time Distribution: 1.0 hr Lec

Expected out-of-class hours per week: 2.0

Graded Component: Lecture

Credit by Examination: No

Admission Condition: No

Off Campus: Never

Campus Locations: Columbus

Prerequisites and Co-requisites: CSE 2231

Exclusions: Not open to students with credit for CSE 459.22

Cross-Listings:

The course is required for this unit's degrees, majors, and/or minors: No

The course is a GEC: No

The course is an elective (for this or other units) or is a service course for other units: Yes

Subject/CIP Code: 14.0901

Subsidy Level: Baccalaureate Course

Programs

Abbreviation	Description
BS CSE	BS Computer Science and Engineering

Course Goals

Be competent with using C++ classes, member functions, constructors, destructors, etc.
Be competent with using templates and the C++ standard template library (STL).
Be competent with using inheritance including using virtual functions.
Be familiar with using arrays and pointers to work with collections of objects and with allocating and releasing memory.
Be familiar with using .h and .cpp files to organize large programs.
Be familiar with the relation between the runtime picture and the source-level picture of moderately complex programs; and using this to build reliable programs.

Course Topics

Topic	Lec	Rec	Lab	Cli	IS	Sem	FE	Wor
Introduction; Compiling and running C++ programs (including use of .h and .cpp files);	0.5							
Overview of simple types in C++; arrays, classes, address types (pointers and references);	1.5							
Distinction between objects and classes; compile-time picture vs. runtime; member functions are invoked on objects; exception: static members (both data and functions);	2.0							
Stack vs. heap objects; automatic vs. explicit creation; constructors and destructors; new and delete; "this" pointer; complex structures such as trees;	3.0							
Inheritance; public vs. private vs. protected; overriding; virtual methods, pure virtual methods, abstract classes; virtual vs. non-virtual methods; runtime dispatch and how it works;	3.0							
Exceptions, namespaces;	2.0							
Templates; examples using STL.	2.0							

Representative Assignments

Programming assignment involving a few simple classes such as simple bank accounts;
Programming assignment involving the use of non-trivial constructors, destructors;
Programming assignment involving pointers, new, delete, "this" variable, complex structures such as trees;
Programming assignment involving inheritance, virtual methods;

Grades

Aspect	Percent
Class attendance, participation	15%
Programming assignments	85%

Representative Textbooks and Other Course Materials

Title	Author
<i>The C++ Programming Language</i>	B. Stroustrup

ABET-EAC Criterion 3 Outcomes

Course Contribution		College Outcome
*	a	An ability to apply knowledge of mathematics, science, and engineering.
*	b	An ability to design and conduct experiments, as well as to analyze and interpret data.
**	c	An ability to design a system, component, or process to meet desired needs.
	d	An ability to function on multi-disciplinary teams.
	e	An ability to identify, formulate, and solve engineering problems.
	f	An understanding of professional and ethical responsibility.
	g	An ability to communicate effectively.

Course Contribution		College Outcome
	h	The broad education necessary to understand the impact of engineering solutions in a global and societal context.
**	i	A recognition of the need for, and an ability to engage in life-long learning.
	j	A knowledge of contemporary issues.
***	k	An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

BS CSE Program Outcomes

Course Contribution		Program Outcome
*	a	an ability to apply knowledge of computing, mathematics including discrete mathematics as well as probability and statistics, science, and engineering;
*	b	an ability to design and conduct experiments, as well as to analyze and interpret data;
**	c	an ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints such as memory, runtime efficiency, as well as appropriate constraints related to economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability considerations;
	d	an ability to function on multi-disciplinary teams;
	e	an ability to identify, formulate, and solve engineering problems;
	f	an understanding of professional, ethical, legal, security and social issues and responsibilities;
	g	an ability to communicate effectively with a range of audiences;
	h	an ability to analyze the local and global impact of computing on individuals, organizations, and society;
**	i	a recognition of the need for, and an ability to engage in life-long learning and continuing professional development;
	j	a knowledge of contemporary issues;
***	k	an ability to use the techniques, skills, and modern engineering tools necessary for practice as a CSE professional;
**	l	an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
*	m	an ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
***	n	an ability to apply design and development principles in the construction of software systems of varying complexity.

Prepared by: Bruce Weide