

CIS2541 – C++ Language Programming

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello
    << "world!" << endl;
    return 0;
}
```

Instructor: Carolyn England

Index

This syllabus includes the following sections.

- [Course Details](#)
- [Course Objectives](#)
- [Course Description](#)
- [Course Materials](#)
- [Due Dates](#)
- [Programming Project](#)
- [Grading Policy](#)
- [Incomplete Policy](#)
- [Satisfactory/Fail \(S/F\) Grade Option](#)
- [Withdrawal Policy](#)
- [Academic Integrity](#)
- [Student Email Accounts](#)

Course Details

Title	CIS 2541 – C++ Language Programming (4 semester hours)
Instructor(s)	Carolyn England
Office Hours	http://www.cod.edu/people/faculty/england
Location	This course is held online. E-mail: england@cod.edu Office: OCC 126A Phone: 630-942-4125
Day and Time	Online (Section NET91)
Textbook(s)	Starting Out with C++: From Control Structures through Objects 7/e by Tony Gaddis, Publisher: Addison Wesley Higher Education, ISBN-10: 0-13-25762502, ISBN-13: 978-0-13-257625-3.
Prerequisite(s)	CIS 1400 – Programming and Logic Technique OR consent of instructor

Starting Date	01/12/2012
Ending Date	05/11/2012

[Index](#)

Course Objectives

Upon successful completion of this course, the student will be able to:

1. Explain the process of compiling and executing C++ language programs
2. Describe C++ language data types, operators, expressions, and flow control structures
3. Demonstrate the use of C++ functions, arrays, strings, and classes
4. Demonstrate the ability to develop interactive procedural and object oriented applications
5. Explain the concept of inheritance, polymorphism, and virtual functions
6. Describe file input and output

[Index](#)

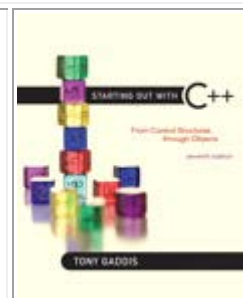
Course Description

[Computer Information Systems 2541](#): Introduces C++ Language Programming, an object oriented programming language. Includes C++ data types, operators, expressions, control structures, functions, arrays, pointers, strings, Abstract Data Types (ADTs), classes, inheritance, polymorphism, virtual functions, and file input/output. Emphasis on building the foundation necessary to understand the capabilities of the C++ programming language and the skills to develop practical procedural and object oriented applications.

[Index](#)

Course Materials

Starting Out with C++: From Control Structures through Objects 7/e by Tony Gaddis, Publisher: Addison Wesley Higher Education, ISBN-10: 0-13-25762502, ISBN-13: 978-0-13-257625-3.



<p>Textbook Companion Website http://www.pearsonhighered.com/gaddis/ (click on "Companion Website" for course text) (access Video Notes, Source Code, and Appendices)</p>
<p>Visual Studio Express Edition Download http://www.microsoft.com/express/Downloads/ (freely available C++ IDE to complete programming assignments)</p>
<p>Adobe Reader Download http://get.adobe.com/reader/ (needed to complete assignment forms)</p>
<p>Adobe Flash Player Download http://get.adobe.com/flashplayer/ (needed for viewing textbook VideoNotes)</p>

[Index](#)

Due Dates

To ensure that you are able to successfully complete this course on a timely basis, various deadlines have been set. The deadlines set forth below are **firm deadlines**, which will not be extended *except in extraordinary circumstance*. (Please note that being busy with work, family, etc. does NOT constitute an extraordinary circumstance.)

Use the Blackboard Course Information Link to access the individual units. Except for assessments, all components of a unit are available on the first day of the semester. **Assessments are available for a limited time before and after the Assignment Due Dates.**

The wise student will submit Assignments earlier than the due date (and take Assessments earlier than the due date). **Discussion Board postings/responses must occur **before 11:50 pm two calendar days prior to the due date** in order to get full credit for that part of the assignment.** All grading for Assignments and Assessments will commence **after** the assigned due dates. A minimum of two week turnaround for grading of Assignments and Assessments can be expected.

Unit	Due Dates	
Unit 0 -- Introduction	Assignment	Monday, January 23, 2012
	Assessment	Wednesday, January 25, 2012
Unit 1 – Computers and Programming	Assignment	Monday, January 30, 2012
	Assessment	Wednesday, February 1, 2012

Unit 2 – Simple Data Types, Expressions, and Equations	Assignment	Monday, February 6, 2012
	Assessment	Wednesday, February 8, 2012
Unit 3 – Selection Control Structures	Assignment	Monday, February 13, 2012
	Assessment	Wednesday, February 15, 2012
Unit 4 – Data Input/Output	Assignment	Monday, February 20, 2012
	Assessment	Wednesday, February 22, 2012
Unit 5 – Looping Logic Structures	Assignment	Monday, February 27, 2012
	Assessment	Wednesday, February 29, 2012
Unit 6 – Arrays	Assignment	Monday, March 5, 2012
	Assessment	Wednesday, March 7, 2012
Unit 7 – Modularity	Assignment	Monday, March 12, 2012
	Assessment	Wednesday, March 14, 2012
Unit 8 – Advanced Array Processing	Assignment	Monday, March 19, 2012
	Assessment	Wednesday, March 21, 2012
Unit 9 – Address Data Types	Assignment	Monday, March 26, 2012
	Assessment	Wednesday, March 28, 2012
Unit 10 – Structured Data Types (<i>Spring Break April 2-8, 2012</i>)	Assignment	Monday, April 9, 2012
	Assessment	Wednesday, April 11, 2012
Unit 11 – Namespaces	Assignment	Monday, April 14, 2012
	Assessment	Wednesday, April 18, 2012
Unit 12 – Object Oriented Programming	Assignment	Monday, April 23, 2012
	Assessment	Wednesday, April 25, 2012
Programming Project	Wednesday, May 9, 2012	
Withdrawal From Class	Tuesday, March 20, 2012	
Request S/F Grading Option	Tuesday, March 20, 2012	
Request Incomplete	Friday, April 27, 2012	

[Index](#)

Programming Project

Each student will be complete a final programming project that combines the concepts learned in each of the units into a single programming exercise. Unit assignments will serve as the practice exercises for each of the unit concepts.

[Index](#)

Grading Policy

Assignment Requirements

Each unit contains an assignment. The assignment is a fill-in pdf form that you can use the freely downloadable Adobe Reader 7.0 or greater to complete. To achieve the maximum points for any unit assignment, the following requirements should be followed:

1. Assignments should be completed by downloading the pdf form for the appropriate unit, completing the questions in the given spaces, saving the changes, and then uploading the completed pdf form.
2. Assignment should include the student's name in the appropriate space at the beginning of the pdf form.
3. The pdf form should be completed in the following manner:
 - check the 'Completed' checkbox for any question that requires completion of an activity but no direct written response in the assignment form
 - complete each text response question in the provided textbox; multi-line textboxes are provided for longer answers
 - if online or other resources are being referenced, the statements **must** be summarized in the student's own words, not copied from the given source (see Academic Integrity)
 - complete and grammatically correct sentences should be used
 - no text-message or e-mail abbreviation format should be used
 - limit responses to no more than 3 paragraphs per question (for the more detailed and complex questions—most will be less than that)
 - indicate name of C++ source code file that will be attached to assignment submission for questions requiring a C++ program
 - indicate name of file containing sample code execution session for questions requiring a C++ program (*this can be the source code file, if included as multi-line comment*)
4. The pdf form should be saved with the student's initials (remember to keep the pdf extension). For example, a student by the name of Peter John Perfect would use the following name for unit assignment number 1:

PJPUnt1.pdf

Make sure your answers are saved and viewable by Adobe Reader before submitting the form. If the form answers cannot be viewed by Adobe Reader, they will not be graded and the student will receive a grade of 0 for the assignment. Completion of the assignment for with a product other than Adobe Reader often leads to answers not being viewable with Adobe Reader. To check if answers are viewable with

Adobe Reader, right-click the pdf assignment file icon and select “Open with” to select Adobe Reader and verify your completed selections.

5. Postings for an assignment should in the appropriate section of the Discussion Board and include the students initials, question number, and short description as the subject. For example, a posting made by Peter John Perfect for question 2 from the assignment of unit 0 would be made under the Discussion Board Forum titled “Unit #0 – Introduction” and have the following subject:

PJPQ2 – Intro

6. Any C++ source code files created for completion of an assignment should be included with the pdf form for assignment submission. These files should be ascii text files with the cpp file extension and **should not include the pre-compiled header preprocessor directive stdafx.h**. Each completed C++ program should be in a single separate file that includes a file header (multi-line comment with file name, author, date, description). The name of all submitted source code file(s) should be identified in the pdf form next to the corresponding assignment question/task. Only a single source file will be graded for each assignment question/task; multiple source file submissions for the same assignment question/task are not allowed and will not be graded. **Source files that cannot be compiled will receive a maximum of one half the possible points for that part of the assignment.**

The output from a sample execution of the C++ source code file(s) should be included as either:

- multi-line comment within source code file,
 - additional ASCII text file, or
 - additional bitmap image file.
7. Be sure to include any needed data files used (these files should also be in ascii text format). All assignment files (source, data, output, pdf form) should be bundled into a **single .zip file** for final assignment submission. A video on creating .zip files in WinXP (which also applies for Windows 7) can be found under “External Links”. **Files bundled in a format other than .zip will not be graded.**
 8. **Discussion Board postings/responses must occur before 11:50 pm two calendar days prior to the due date in order to get full credit for that part of the assignment.** Assignments should be submitted by the assigned due dates in order to get full credit. **Even if the assignment does not require text responses within the pdf form (only the ‘Completed’ checkboxes), the student must still upload the completed pdf form and select the “Submit” button to indicate completion of the assignment requirements by the specified due date. Assignments that have not been “Submitted” with a completed**

pdf form by the assignment due date will receive a grade of 0 for that assignment.

9. The student is encouraged to get clarification and assistance with any difficulties related to the assignment. There are several options available to the student:
 - classroom discussion boards for each unit
 - direct contact with the instructor via e-mail (allow 24 hour turnaround for question response during the Monday to Sunday week--weekend questions will be responded to on Monday)

Programming Project Requirements

This course requires the completion of a programming project that combines the concepts learning in each of the unit assignments into a single programming project. To achieve the maximum points for this programming project, the following requirements should be followed:

1. Project should be completed by downloading the pdf form for the programming project, completing the questions in the given spaces, saving the changes, and then uploading the completed pdf form.
2. Project should include the student's name in the appropriate space at the beginning of the pdf form.
3. The pdf form should be completed in the following manner:
 - check the 'Completed' checkbox for any question that requires completion of an activity but no direct written response in the project form
 - complete each text response question in the provided textbox; multi-line textboxes are provided for longer answers
 - if online or other resources are being referenced, the statements **must** be summarized in the student's own words, not copied from the given source (see Academic Integrity)
 - complete and grammatically correct sentences should be used
 - no text-message or e-mail abbreviation format should be used
 - limit responses to no more than 3 paragraphs per question (for the more detailed and complex questions—most will be less than that)
 - indicate name of attached files containing:
 - data input for project,
 - C++ source code
 - sample project run, and
 - data output for project.
4. The pdf form should be saved with the student's initials (remember to keep the pdf extension). For example, a student by the name of Peter John Perfect would use the following name for the project assignment:

Make sure your answers are saved and viewable by Adobe Reader before submitting the form. If the form answers cannot be viewed by Adobe Reader, they will not be graded and the student will receive a grade of 0 for the assignment. Completion of the assignment for with a product other than Adobe Reader often leads to answers not being viewable with Adobe Reader. To check if answers are viewable with Adobe Reader, right-click the pdf assignment file icon and select "Open with" to select Adobe Reader and verify your completed selections.

5. The C++ source code file(s) created for completion of the project should be included with the pdf form for project submission. These files should be ascii text files with the cpp file extension. Only a single version of the project file(s) will be allowed; multiple project file submissions for the same project are not allowed and will not be graded. **Source files that cannot be compiled will receive a maximum of three quarters (75%) the possible points for the programming project assignment.**
6. Be sure to include any data files used as well as a sample of generated output (these files should also be in ascii text format). All assignment files (source, data, output, pdf form) should be bundled into a single .zip file for final submission. A video on creating .zip files in WinXP (which also applies for Windows 7) can be found under "External Links". **Files bundled in a format other than .zip will not be graded.**
7. Project should be submitted by the assigned due date. **Be sure to upload the completed pdf form with the C++ source code. Click the "Submit" button to indicate completion of the project requirements by the specified due date. Projects that have not been "Submitted" with a completed pdf form and accompanying C++ source code by the project due date will receive a grade of 0 for the programming project.**
8. The student is encouraged to get clarification and assistance with any difficulties related to the project. There are several options available to the student:
 - classroom discussion boards for project
 - direct contact with the instructor via e-mail (allow 24 hour turnaround for question response during the Monday to Sunday week--weekend questions will be responded to on Monday)

Assessment Requirements

Each unit contains an assessment. To achieve the maximum points for any unit assessment, the following guidelines should be applied:

1. **Allow yourself enough time to complete the assessment in one sitting.** Assessments should be completed within the designated time limits. An overage of 1 or 2 minutes is allowed. Completion times greater than 5 minutes over the designated time limit will have points deducted from the final score (i.e. **10% of total possible assessment points for each 5 minute overage**).
2. Click the "Save" button periodically to save your responses should you accidentally disconnect from Blackboard. This will allow some of your answers to be recorded.
3. Click the next button (a right arrow) to advance to the next question. **Once you have advanced to the next question, you cannot return to a previous question.** Click the "Submit" button when you have completed the assessment. **Assessments that have not been "Submitted" by the due date will receive a grade of 0 for the assessment.**
4. The assessment clock runs from the time you first select "Start" to the time you click "Submit", regardless of whether the assessment is visible in your Browser. This implies that the **assessment clock does not stop running** if you decide to quit and come back later to complete the assessment.

Points

Points will be distributed in the following manner:

Assignments	530
Assessments	370
Program	100
Total	1000

Final Grade

Final Grades will be assigned using the following scale:

Accumulated Points	Grade	Percentage
900 – 1000	A	≥ 90
800 – 899.9	B	80 – 89
700 – 799.9	C	70 – 79
600 – 699.9	D	60 – 69
599.9 or lower	F	< 60

[Index](#)

Incomplete Policy

In order to request an incomplete grade, you must obtain permission from the instructor, satisfy minimum completion requirements (see below), and complete an Incomplete contract from your student e-mail account by the Due Date. A minimum amount of completed work will be required to request an Incomplete grade. Contact the instructor **before** the semester Due Date if you are interested in receiving an Incomplete grade.

An "I" grade will be given only when **ALL** of the following conditions are met:

1. The student must 'sign' an Incomplete contract on or before the Due Date. No Incomplete grade can be given until the student signs the Incomplete contract. The contract will indicate the work to be completed and the completion date.
2. The student must complete **ALL** work for Units 0 to 10 (Assignments and Assessments) on or before the Due Dates.
3. The student must be working at a "C" level or higher.

All incomplete work must be completed by the contracted date, which is usually around midterm of the following semester. It is the student's responsibility to watch this date carefully. No extensions beyond this date will be given.

The College policy on Incomplete Grades can be found on page 91 of the College catalog under Academic Policies and Procedures:

http://www.cod.edu/catalog/AcademicPolPro_11-13.pdf

[Index](#)

Satisfactory/Fail (S/F) Grade Option

The S/F grade option is available to students in this course. It **may** be issued under the following conditions:

1. A written statement must be completed by the student and the instructor on or before the Due Date. (See Due Dates for current semester Due Date.)
2. The student must have achieved points equivalent to or greater than 75% of the total possible points in order to receive a Satisfactory (S) grade. If a student's points are less than 75%, a grade 'F' will be assigned for the course.
3. Once an S/F grade has been registered with the Records Office, it cannot be changed.

4. An 'S' grade will not be computed in the GPA; the 'F' grade, however, will be used in computing the GPA.

[Index](#)

Withdrawal Policy

- **Course Withdrawals**

You are encouraged to consult directly with the Instructor when considering a course withdrawal. You may withdraw from a course through myAccess or in person at the Registration office on or before the date set forth in Due Dates. If you withdraw from the course by this date, a withdrawal or 'W' will appear on your transcript. Thereafter, a grade will be assigned which reflects your actual performance in the class. Your instructor does not provide permits to withdraw from class after the official Mid-Term withdrawal date.

- **Medical Withdrawals**

Requests for medical withdrawals should be made to the Dean of Enrollment Services. Verification from a physician or medical institution is typically required.

- **Administrative Withdrawals**

At the discretion of the Instructor, students not actively pursuing the completion of course objectives **may** be withdrawn from the class by the Instructor any time up to two weeks prior to the end of the term and given a grade of 'W.'

No withdrawals will be given after the [Due Date](#).

The College policy on Withdrawals can be found on page 93 of the College catalog under Academic Policies and Procedures:

http://www.cod.edu/catalog/AcademicPolPro_11-13.pdf

[Index](#)

Academic Integrity

Course-Related Academic Integrity is one aspect of the Student Code of Conduct. The Student Code of Conduct can be found on page 98 of the College catalog under Student Services and Information:

http://www.cod.edu/catalog/StudentServices_11-13.pdf

Students violating this policy will be processed as indicated in the Student Code of Conduct.

[Index](#)

Student E-mail Accounts

Much of the correspondence for this course will occur via discussion boards, announcements, and file uploads. However, all COD students are issued a myACCESS user id that gives access to a variety of college information services. One of these services is access to student e-mail. If you have not used myACCESS before, use the following link to get the myACCESS home page:

<https://myaccess.cod.edu>

The following site provides access to a variety of resources on how to get started using myACCESS and Student e-mail.

<http://www.cod.edu/it/labs/pages/emailguide.html>

The student is responsible for periodically monitoring their COD student e-mail account for any course related and/or official communication from the instructor.

[Index](#)
