



Syllabus

SCHOOL OF TECHNOLOGY AND COMPUTING **IS 425: C# - Advanced**

5 Credits
Effective: Spring 2013

Access to the Internet is required.
All written assignments must be in Microsoft-Word-compatible formats.
See the library's APA Style Guide tutorial for a list of resources that can help you use APA style.

FACULTY

Faculty Name: FACULTY NAME

Contact Information: CONTACT INFORMATION

[INSTRUCTOR MAY INSERT PERSONAL MESSAGE IF DESIRED]

COURSE DESCRIPTION

In this course, students use C# to implement applications using the Windows Presentation Foundation (WPF). Students are introduced to WPF and XAML, and then learn to build a WPF application. The course also covers advanced topics in C#. The course prepares students to program in a graphical environment.

COURSE RESOURCES

Required and recommended resources to complete coursework and assignments are available from the [Course Document Lookup](#).

CITYU LEARNING GOALS

This course supports the following City University learning goals:

- Professional competency and professional identity

COURSE OUTCOMES

In this course, learners:

- Bind data elements and collections to interactive interface elements.
- Apply C# to create and manage data queries and query results.
- Create graphical interaction applications.
- Follow quality coding practices to produce documented, legible, maintainable, reusable and efficient code.
- Use XAML and C# to design interactive applications.

CORE CONCEPTS, KNOWLEDGE, AND SKILLS

- Accept user input using Text Controls.
- Access resources in code.
- Add code to handle application events.
- Add resources and retrieve them for use in an application.
- Apply conditional formatting.
- Attach event handlers to styles.
- Bind to a collection of objects.
- Bind to a LINQ expression.
- Bind to objects that aren't elements.
- Build a Data Access Component.

- Build a data object.
- Build localizable interfaces.
- Construct a collection with an initial set of members using a collection initializer.
- Create a coordinate-based layout with the canvas.
- Create an application object.
- Create and use assembly resources.
- Create anonymous and implicit typed variables.
- Create event handlers.
- Create interactions between windows.
- Create layouts using the core Layout Panels.
- Create objects with a Value Converter.
- Declare XAML namespaces as attributes in a document.
- Define a style as a collection of property values to an element.
- Define deferred execution and explain their effects on queries.
- Describe at a high level the architecture of WPF.
- Describe how binding updates the target automatically.
- Describe how C# partial classes are used to merge application code with design code.
- Describe how XAML (extensible Application Markup Language) is used to separate the graphical design and underlying code and enable integration of the workflow between them.
- Describe the difference between bitmap and vector graphics.
- Describe the use of other query operations.
- Detail nesting rules.
- Develop and apply a grid-based layout.
- Diagram the inheritance hierarchy for shapes.
- Diagram the process of compiling and loading XAML.
- Diagram the results of join operations.
- Differentiate between tunneled, bubbling and direct events.
- Discuss focus in WPF.
- Discuss how WPF is supported in Visual Studio.
- Discuss the five key principles that make WPF attractive for developers.
- Display collections of items using list controls.
- Display the bound data object.
- Divide windows into client and non-client areas.
- Draw 2-D graphical content using shapes.
- Evaluate property value inheritance and style inheritance to determine the attributes of an element.
- Examine the base control class and how it supports brushes and fonts.
- Examine the five key categories of WPF events (lifetime, mouse, keyboard, stylus, and multitouch).
- Explain deferred execution with query expressions.
- Explain the concept of resolution independence and demonstrate how to calculate a physical unit size on a display.
- Explain the five available binding modes.
- Explain the purpose and use of headered content controls.
- Explain the role of a type converter.
- Explain the three critical consequences of shapes being elements.
- Explore routed events and how they allow events to be handled at different levels.
- Explore the menu class.

- Explore the Window class.
- Format strings with a Value Converter.
- Format the output of query expressions using grouping and sorting.
- Group results using Groupby().
- Implement commands using the WPF command model.
- Implement date controls.
- Implement various content controls.
- Include a status bar in Windows.
- Include the using declarative for System.Linq to enable use of standard query operators.
- Inject flair into basic elements and standardize the appearance of an application.
- Learn various ways to show and position windows.
- List and explain the basic properties of the Windows class.
- List and explain the four key principles for WPF layout.
- List the services WPF provides for programmers.
- Manipulate the key attributes of layout panels.
- Nest Layout containers.
- Outline the concept of a markup extension.
- Outline the role of DirectX in supporting WPF.
- Perform common application tasks such as show splashscreens, process command-line arguments, and support interaction between users.
- Register file types with windows and associate it with an application.
- Review the technologies on which Windows Presentation Foundation (WPF) is based.
- Set values within a prescribed range using Range-based controls.
- Size, place and scale shapes.
- Transform shapes and elements.
- Understand how you can bind an element to another element.
- Understand the basic ground rules of XAML.
- Understand the infrastructure that underpins the Application class.
- Use a data template for sophisticated visual representations.
- Use an enumerator to iterate through a collection.
- Use brushes to fill areas.
- Use context and separators to format menus.
- Use MenuItem to create a menu structure.
- Use paths and geometries to create more complex drawings.
- Use query expressions to filter and project collections.
- Use resources in styles.
- Use standard query operators to search, collect and count items in a collection.
- Use the LocBaml tool to extract items for localization and build satellite assemblies.
- Use the StringFormat property.
- Use the Toolbar class to create and format toolbars.
- Use the visibility property to determine what is seen in an interface.
- Use triggers to automate simple style changes.

OVERVIEW OF COURSE GRADING

The grades earned for the course will be derived using City University of Seattle’s decimal grading system, based on the following:

<i>Overview of Required Assignments</i>	<i>% of Final Grade</i>
Program 1 - Magic 8 Ball	10%
Program 2 - Patents Program	5%
Program 3 - Will Pigs Fly?	10%
Program 4 - Random Facts	10%
Program 5 - Packaged Facts	10%
Program 6 - Data Maintenance Tool	10%
Program 7 - Inject Flair	10%
Program 8 - Shapes, paints and scribbles.	5%
Program 9 - Non-rectangularities	5%
Program 10 - Menus and Tool bars	5%
Instructor Determined Assignments (also including participation)	20%
TOTAL	100%

SPECIFICS OF COURSE ASSIGNMENTS

The instructor will provide grading rubrics that will provide more detail as to how this assignment will be graded.

Program 1 - Magic 8 Ball

Students will create a version of the Magic 8 Ball program (or download it from the book website) and change the design and layout from the original. Students will move the declaration of answers inside the GetRandomAnswer function and make it an anonymous type “var answers = “. Add additional buttons to give answers of a different character – for example: pirate, pessimist, valley girl, etc. Change the manner in which the return string is selected to use the input in some manner to calculate the response – for example: hash the input to determine which answer to select, select an answer which begins with the 3rd letter of the input, or reorder the words of the input and add negative or positive words (e.g. “will I get lunch” becomes “lunch no will I get” or “lunch yes I will get”).

Students will publish the application and create a report on the application including a description and screen shots to demonstrate its operation. The report will also include any significant source files, or source snippets. Students will submit the report to the instructor and make the application available for installation.

<i>Components</i>	<i>% of Grade</i>
XAML and C#	50%
Quality coding practices	30%
Requirements	20%
TOTAL	100%

Program 2 - Patents Program

Students will create a graphical interface for the patents classes (Michaelis and Lippert pp. 578-580). The interface will display the contents of the collections in different areas of the interface. The layout should include at least two different panel types. There should also be buttons to provide filtered lists and counts of items in the collections. The students will experiment with the layout.

Upon completion of the program students will publish the application and create a report on the application including a description and screen shots to demonstrate its operation. The report should demonstrate at least 3 layout alternatives considered in determining the final layout. The report must explain why the panel types were chosen over alternatives. It should also include any significant source/configuration files or snippets. The students will submit the report to the instructor and make the application available for installation.

<i>Components</i>	<i>% of Grade</i>
Graphical applications	50%
Data Queries and Results	30%
Quality coding practices	20%
TOTAL	100%

Program 3 - Will Pigs Fly?

Students will merge the “Magic 8 Ball” and “Patents” applications together. The merged application will respond to inquiries with sets of grouped and sorted facts from the patents base made into obscure statements created using query operations. For example if a user enters “Will pigs fly?” the program would respond “Did you know Fred Flintstone invented the harmonica? And Will Rogers invented the accordion?” or some other obscure response.

The application will respond on events such as mouse clicks or keyboard events. Students can add elements that can generate additional events. At least 4 events need to be handled.

Upon completion of the program students will publish the application and create a report on the application including a description and screen shots to demonstrate its operation. The report should also, include any significant source/configuration files or snippets. The students will submit the report to the instructor and make the application available for installation.

<i>Components</i>	<i>% of Grade</i>
Data Queries and Results	30%
Quality coding practices	20%
Graphical applications	50%
TOTAL	100%

Program 4 - Random Facts

Students will continue to expand the “Random Facts” program using the controls and techniques to add functionality and improve the look and feel of the interface. The application should implement and demonstrate at least five techniques from this module.

Upon completion of the program students will publish the application and create a report on the application including a description and screen shots to demonstrate its operation. The report should also include any significant source/configuration files or snippets. The students will submit the report to the instructor and make the application available for installation.

<i>Components</i>	<i>% of Grade</i>
Graphical applications	50%
Data Queries and Results	30%
Quality coding practices	20%
TOTAL	100%

Program 5 - Packaged Facts

Students will package the “Random Facts” program into a more formal model. The packaged program will include localization and the ability to support alternative data sets. The choice of locality and data set will be a command line option. The datasets should be read from a file of a file type that the user registers with Windows. The application will include a fancy splash screen and resources that are used in the application.

Upon completion of the program students will publish the application and create a report on the application including a description and screen shots to demonstrate its operation. The report should also include any significant source/configuration files or snippets. The students will submit the report to the instructor and make the application available for installation.

<i>Components</i>	<i>% of Grade</i>
Graphical applications	50%
XAML and C#	30%
Quality coding practices	20%
TOTAL	100%

Program 6 - Data Maintenance Tool

Students will use bound data objects and elements to develop a data maintenance tool for the “Random Facts” application.

Upon completion of the program students will publish the application and create a report on the application including a description and screen shots to demonstrate its operation. The report should also include any significant source/configuration files or snippets. The students will submit the report to the instructor and make the application available for installation.

<i>Components</i>	<i>% of Grade</i>
Bind data	50%
XAML and C#	30%
Quality coding practices	20%
TOTAL	100%

Program 7 - Inject Flair

Students will apply the tools and techniques of this module to inject flair into their “Random Facts” application. The improvement must include at least 5 properly applied tools and techniques. Students are encouraged to include up to 37 pieces of flair.

Upon completion of the program students will publish the application and create a report on the application including a description and screen shots to demonstrate its operation. The report should also include any significant source/configuration files or snippets. The students will submit the report to the instructor and make the application available for installation.

<i>Components</i>	<i>% of Grade</i>
Graphical applications	50%
XAML and C#	30%
Quality coding practices	20%
TOTAL	100%

Program 8 - Shapes, paints and scribbles.

Students will apply the tools and techniques of this module to draw shapes, paints and scribbles to improve the “look and feel” of their “Random Facts” application. The improvement must include at least 5 properly applied tools and techniques.

Upon completion of the program students will publish the application and create a report on the application including a description and screen shots to demonstrate its operation. The report should also include any significant source/configuration files or snippets. The students will submit the report to the instructor and make the application available for installation.

<i>Components</i>	<i>% of Grade</i>
Graphical applications	50%
XAML and C#	30%
Quality coding practices	20%
TOTAL	100%

Program 9 - Non-rectangularities

Students will create a non-rectangular window for their “Random Facts” application, add a custom icon to the task bar and demonstrate the use of at least 5 basic Windows class properties.

Upon completion of the program students will publish the application and create a report on the application including a description and screen shots to demonstrate its operation. The report should also include any significant source/configuration files or snippets. The students will submit the report to the instructor and make the application available for installation.

<i>Components</i>	<i>% of Grade</i>
Graphical applications	50%
XAML and C#	30%
Quality coding practices	20%
TOTAL	100%

Program 10 - Menus and Tool bars

Students will add menus and toolbars to their “Random Facts” applications. The menus and toolbars need to include at least 5 choices.

Upon completion of the program students will publish the application and create a report on the application including a description and screen shots to demonstrate its operation. The report should also include any significant source/configuration files or snippets. The students will submit the report to the instructor and make the application available for installation.

<i>Components</i>	<i>% of Grade</i>
Graphical applications	50%
XAML and C#	30%
Quality coding practices	20%
TOTAL	100%

Instructor Determined Assignments (also including participation)

Students will participate in activities as defined by the instructor. Whether in class, online, or in a mixed mode setting, students will be graded on their participation in discussions. Student will be graded on their ability to present, explain or defend alternative viewpoints; and the degree to which they have mastered the concepts and principles inherent in the course. Other instructor determined assignments may include, but are not limited to research tasks, outlines, and drafts of required assessments, peer reviews, journals, and readings. Written work will be assessed not only on relevance to the subject presented, but also on adherence to good written form and professional presentation. The instructor may modify the grading components for his or her assignments as appropriate.

<i>Components</i>	<i>% of Grade</i>
XAML and C#	20%
Bind data	20%
Data Queries and Results	20%
Graphical applications	20%
Quality coding practices	20%
TOTAL	100%

COURSE POLICIES

Late Assignments

LATE ASSIGNMENT

Participation

PARTICIPATION

Professional Writing

Assignments require error-free writing that uses standard English conventions and logical flow of organization to address topics clearly, completely, and concisely. CityU requires the use of APA style.

UNIVERSITY POLICIES

You are responsible for understanding and adhering to all of City University of Seattle's academic policies. The most current versions of these policies can be found in the [University Catalog](#) that is linked from the CityU Web site.

Scholastic Honesty

Scholastic honesty in students requires the pursuit of scholarly activity that is free from fraud, deception and unauthorized collaboration with other individuals. You are responsible for understanding CityU's policy on scholastic honesty and adhering to its standards in meeting all course requirements. A complete copy of this policy can be found in the [University Catalog](#) in the section titled *Scholastic Honesty* under *Student Rights & Responsibilities*.

Attendance

Students taking courses in any format at the University are expected to be diligent in their studies and to attend class regularly.

Regular class attendance is important in achieving learning outcomes in the course and may be a valid consideration in determining the final grade. For classes where a physical presence is required, a student has attended if s/he is present at any time during the class session. For online classes, a student has attended if s/he has posted or submitted an assignment. A complete copy of this policy can be found in the [University Catalog](#) in the section titled *Attendance Policy for Mixed Mode, Online and Correspondence Courses*.

SUPPORT SERVICES

Disability Resources

If you are a student with a disability and you require an accommodation, please contact the Disability Resource Office as soon as possible. For additional information, please see the section in the [University Catalog](#) titled *Students with Special Needs* under *Student Rights & Responsibilities*.

Library Services

CityU librarians are available to help you find the resources and information you need to succeed in this course. Contact a CityU librarian through the [Ask a Librarian](#) service, or access [library resources and services online](#), 24 hours a day, seven days a week.

Smarthinking

As a CityU student, you have access to 10 free hours of online tutoring offered through Smarthinking, including writing support, from certified tutors 24 hours a day, seven days a week. Contact CityU's Student Support Center at help@cityu.edu to request your user name and password.