



Object Oriented Programming

Custom Program

Overview

Deadline: Week 13, 5 June 2025, 23:59:00 (firmed)

At this stage, you should have a conceptual understanding of OOP principles and be familiar with the development of SplashKit and SwinAdventure programs. Now it's time to start thinking about creating your own custom program.

There are four different custom program options, each with its own set of learning outcomes, engagement requirements, marking criteria, and grade allocation. You only need to choose one of the options. Your choice should be based on your OOP learning goals and workload balance.

We strongly recommend that you consult with your allocated lab instructor before selecting an option.

Learning Outcomes

Option 1. Fundamental - up to 15 marks. The Unit Learning Outcomes (ULOs) of this option include:

- 1.1 Explaining the principles of object-oriented programming (OOP), including abstraction, encapsulation, inheritance, and polymorphism.
- 1.2. Using OOP and associated .NET and SplashKit class libraries, including file content manipulation, user keyboard interaction, and shape drawing classes.
- 1.3. Developing and testing a custom SplashKit program using Visual Studio Code.

Option 2. Intermediate -- up to 18 marks. This option is an extension of the previous Option 1. In addition to the ULOs of Option 1, it also includes the following ULOs:

- 2.1. Design new drawing classes that respond to various user keyboard interactions.
- 2.2. Construct appropriate UML class and sequence diagrams to communicate and describe your custom project.

Option 3. Advanced - up to 21 marks. In this option, you are required to plan and outline your own custom program. You will then implement it, deliver a presentation, participate in an interview, and write a report about your program.

At the end of this document, a list of suggested custom program ideas is provided. However, you are encouraged to choose an exciting and unique idea that aligns with your future career interests.

- 3.1. Describe and explain the factors that contribute to a good OO programming practices and reflecting your own experiences.

Option 4. Advanced - up to 30 marks. This option builds upon Option 3. In addition to the ULOs of Option 3, it includes the following additional ULO:

- 4.1. Describe and explain the factors that contribute to effective object-oriented software design patterns, reflecting on your own experiences and drawing upon accepted best practices.

Engagement and submission requirement

Option 1 and Option 2.

+ You must have at least one in-person check-in discussion with your allocated lab instructor during Week 9, Week 10, or Week 11—either in a helpdesk session or during lab time. This check-in is to discuss your implementation progress and receive feedback.

+ Failure to complete the check-in will result in an interview with a panel in Week 14. The panel will determine your grade based on your interview performance.

+ Final submission, including source code and a video demonstration, is due in Week 13. If you are completing Option 2, you must also submit a report containing your UML class diagram and at least one sequence diagram describing your implementation (**see more details in pages 4 and 5**).

Option 3 and Option 4.

+ You must have at least one in-person check-in discussion with your allocated lab instructor during Week 9, Week 10, or Week 11—either during helpdesk sessions or in the lab. This check-in is for discussing your implementation, reporting progress, and receiving feedback.

+ **A milestone submission is due in Week 10.** It should include a basic overview of your program (template provided), a class diagram (photo or scanned copy), and at least one sequence diagram (photo or scanned copy)

+ Final submission, including source code and a report, is due in Week 13 (**see more details in pages 6-8**).

+ An interview with a panel will be conducted in Week 14.

Marking Criteria

Option 1. Marks will be based on the number of features implemented, the correctness of the source code, and the clarity of the video demonstration.

Note. *Failure to submit both the source code and video demonstration will result in a mark of zero.*

Option 2. Marks will be based on the number of features implemented, the correctness of the source code, the clarity of the video demonstration, and the quality of the report.

Note: *Failure to submit both the source code and video demonstration will result in a mark of zero.*

Option 3. Marks will be based on the complexity of your custom program, the correctness of the source code, the quality of the report, and your performance in the interview.

Note: *Failure to attend the interview will result in your submission being assessed as an Option 2 project.*

Failure to submit the check-in progress in Week 10 will result in your submission being assessed as an Option 2 project.

Option 4. Marks will be based on the complexity of your custom program, the number of design patterns used, the correctness of the source code, the quality of the report, and your performance in the interview.

Note: *Failure to attend the interview will result in your submission being assessed as an Option 2 project.*

Failure to submit the check-in progress in Week 10 will result in your submission being assessed as an Option 2 project.

****Regarding Options 3 and 4, we will be holding interview sessions in Week 14.** The interview timetable will be announced in Week 12, and slots will be allocated on a first-come, first-served basis.

Each interview will be approximately 10–15 minutes long. You should prepare a short presentation slide highlighting the key features of your custom program. The interview will include a brief live demonstration, followed by several questions from the panel.

Specifications

I. Option 1. Fundamental custom program (15 marks)

This program continues the ShapeDrawing program that you developed in Week 7 by adding following three features.

1. Allow the end-user to press a key that will simultaneously draw a random number of different shape types—including rectangles, circles, and lines—onto the canvas. You may choose which key to use. The shapes should appear at random locations and have random colors. Other attributes such as width and height (for rectangles), radius (for circles), or length (for lines) can be hard-coded.
2. Allow the end-user to press a key that will automatically **draw the first letter of your first name on the canvas**. You may choose which key to use, but it must be different from any keys already used in your existing ShapeDrawing program. You may use lines, rectangles, circles, or a combination of these shapes to form the letter in a clear and visible way.
3. Allow the end-user to press a key that will automatically change the colors of all shapes on the canvas to random colors.

While developing the three new features, you must ensure that the existing functionalities from Week 7 are preserved—specifically, the ability to save and load shapes to and from a file.

After completing your implementation, create a video recording to demonstrate the features you have developed and explain your source code. At the beginning of the video, please clearly state your name and student ID.

Submission Details:

+ Source Code

+ Video Recording for Demonstration

The submission deadline is Week 13.

The submission link will be available on Canvas under Assignment section.

Noting that attempting Option 1 does not mean you will get up to 15 marks by default. It depends on your source code correctness, video demonstration performance and academic integrity.

Specifications

II. Option 2. Intermediate custom program (18 marks)

This program continues the ShapeDrawing program that you have developed in Week 7, adding following three features.

1. Allow the end-user to press a key that will simultaneously draw a random number of different shape types—such as rectangles, circles, and lines—onto the canvas. You may choose which key to use. The shapes should appear at random locations and have random colors. Other attributes—such as width and height for rectangles, radius for circles, and length for lines—can be hard-coded.
2. Allow the end-user to press a key that will automatically **draw your first name on the canvas**. You may choose which key to use, but it must be different from any keys used in your existing ShapeDrawing program. You can use lines, rectangles, circles, or a combination of these shapes to make your name clear and visually distinct.
3. Allow the end-user to press a **keyboard that will automatically scale down the size** of all available shapes on canvas. For example, reducing the width and height, radius, or length.

While developing the above three features, you must ensure that the existing functionalities from Week 7 are preserved—specifically, saving and loading shapes to and from a file.

After completing the implementation, create a video recording to demonstrate the functionalities you have developed and explain your source code. At the beginning of the video, please clearly display your name and student ID.

You are also required to prepare a report that includes a UML class diagram and a sequence diagram. These diagrams should illustrate how objects collaborate and interact with each other to achieve either the second or the third feature.

Submission Details:

+ Source Code

+ Demonstration Recording

+ A UML report

The submission deadline is Week 13.

The submission link will be available on Canvas under Assignment section.

Noting that attempting Option 2 does not mean you will get 15-18 marks by default. It depends on your source code correctness, video demonstration performance, the report's quality and academic integrity.

Specifications

III. Option 3. Advanced custom program (21 marks)

Prior choosing this task, we strongly recommend you to discuss with your allocated lab instructor. This project is your own custom program. In this task, you will provide a plan and overview of the structure of a custom program (something you would be interested in creating). Attached to this specification, we provide the list of potential program ideas. Your custom program should:

1. Demonstrate the use of abstraction -- create your own classes that model the domain.
2. Demonstrate the use of inheritance and polymorphism
3. Demonstrate the use of UML class diagrams to explain how your solution works.
4. Provide supporting arguments and real-world evidence to show that your custom program is practical and useful to its target community and end-users

Here are some steps to get you started.

1. Download the **Design Report** template.
2. Provide a summary of your program — What does it do? What are some of the key features etc.
3. Describe the main roles: enumerations, classes & interfaces.
4. Describe the main responsibilities for the classes and interfaces. Get some detail down now for your tutor to check, but there is no need to spend ages on this task. Have enough that you can start to see how the program will continue to develop as you proceed.
5. Show your plans to your tutor, lecturer, help desk staffers, and/or friends to get some feedback.

Submission Details

+ **One milestone submission in Week 10** to report:

- a basic overview of your program (template is provided) and
- class diagram (photo or scan), and a picture of one or more sequence diagram (photo or scan).

Noting that, this milestone is compulsory. Your tutor will provide feedback on it.

- + Final submission in Week 13 including source code and final report. The report include the Design Report with the final UML class diagram, usage document outlining what your program does and how it works with screenshots, and a short summary of what you have learned from your custom program.
- + Prepare slide presentation and conduct interview in Week 14.
- + Noting that attempting Option 3 does not mean you will get 18-21 marks by default. It depends on your custom program's quality and complexity, report quality, interview performance, and academic integrity.

Specifications

III. Option 4. Challenging custom program (30 marks)

Prior choosing this task, we strongly recommend you to discuss with your allocated lab instructor. This project is your own custom program. It is an extension of the Option 3. Attached to this specification, we provide the list of potential program ideas. Using the Option 3 design as a starting point, add details on:

1. Which design patterns you plan on using (at least 3 design patterns).
2. What complexity or additional functionalities you will add to elevate your program from the option 3 to this level.
3. Why is it not simple compared to the option 3.
4. You should update your UML class diagram where appropriate.

Here are some steps to get you started.

1. Download the **Design Report** template.
2. Provide a summary of your program — What does it do? What are some of the key features etc.
3. Describe the main roles: enumerations, classes & interfaces.
4. Describe the main responsibilities for the classes and interfaces. Get some detail down now for your tutor to check, but there is no need to spend ages on this task. Have enough that you can start to see how the program will continue to develop as you proceed.
5. Reflect on any additional features or complexity you could add
6. Identify opportunities for applying design patterns, or places where you may have already used a design pattern. We would expect to see at least 2, or 3 if you are using a simpler pattern such as Singleton.
7. Describe the design patterns you will use and why.
8. Update your UML class diagram to include the design patterns you have chosen.
9. Describe what makes your new program design more complex and/or well done than an Advanced level custom program design.
10. Show your plans to your tutor, lecturer, help desk staffers, and/or friends to get some feedback.

Submission Details

+ **One milestone submission in Week 10** to report:

- a basic overview of your program (template is provided) and
- class diagram (photo or scan), and
- a picture of one or more sequence diagram (photo or scan).
- what potential design patterns you plan to use

Noting that, this milestone is compulsory. Your tutor will provide feedback on it.

+ Final submission in Week 13 including source code and final report. The report include the Design Report with the final UML class diagram, usage document outlining what your program does and how it works with screenshots, and a short summary of what you have learned from your custom program.

+ Prepare slide presentation and conduct interview in Week 14.

+ Noting that attempting Option 4 does not mean you will get 21-30 marks by default. It depends on your custom program's quality and complexity, report quality, interview performance, and academic integrity.