WI21 ITGM 220 CORE PRINCIPLE: PROGRAMMING

ASSIGNMENT 05

05. Custom Classes (25%)

Learning Outcomes

After completing this Assignment, students will be able to:

- Model information in custom classes.
- Create applications using objects defined in custom classes.

Key skills:

1. Object-oriented structure including class inheritance Disciplined, documented process of research and development
2. Design and implementation of more complicated systems

Overview

Create an interactive application / game utilizing Object Oriented Programming principles.

Requirements

1. Project **must** have a start screen with the title of the work and creator's name clearly displayed.
2. Project **must** implement at least 1 base class.
3. Project **must** implement at least 1 child class..
4. Project **must** tell a story through user input. Example:
    a. Observe the effects of vaccination by looking at infection rate of polio during the year 1850 - 1970.
    b. Feel the calm of the world by watching the sun set and the moon rises.
    c. Bounce this ping pong ball more times then your opponent.
    d. Shoot down as many aliens as you can before you get hit by friendly fire.
    e. Gobbling up as many apples as you can before you take a bite out of your own growing snake body.

Procedure

1. Research online and come up with an idea for an interactive application or game.
2. Collect a set of image references that will influence your design, create a mood board.
3. Collect assets that you would be adding to your project. (example: png, svg, wav etc)
4. Sketch out the title screen + main screenshot of the project. Evolve your design by sketching out the next version. Do not erase old versions as you would need them to prove the originality of the idea. If you are using external images as elements, re-create the final look and feel either through pencil sketches or image editing software.
5. Based on your sketch, create a plan for your project in processing:
    a. What are the base / child classes? What are the functions in the class?
    b. How many objects would you need?
    c. You would need 1 start screen. How many more screens would you need?
    d. What colors / fonts will you use?
    e. How does the user react to the sketch?
6. Write the code to create your project in Processing.
    a. Stage Size Range: 500 x 500 px (smallest) - 1200 x 800 px(largest)
    b. Double check to see if you have met the requirements as stated in the **Requirements** Section.
    c. Maintain good code structure as shown in class (Bracket Placements, Comments, Headings, Variable / Function / Class Names
7. Capture Output
    a. Record the play session using OBS / Quicktime
    b. Include the video clip in the Project presentation.
8. Create Project Presentation Deck.
    a. Consult Sample Project Documentation Structure Guide Below.

Submission and Due Date

1. Create an assignment report by Exporting a PDF of your presentation deck.
2. Properly ZIP your saved project.
    a. Double Check one last time and make sure the project runs smoothly..
    b. Make sure you have included the complete folder content, along with your assignment report pdf.
    c. Remember the project might NOT execute if missing essential pngs / sound / libraries.
3. Name your submission zip file correctly:

a. WI21_ITGM220_FirstNameLastName_A4datavisualization.zip
b. for example, John Doe would name the file "WI21_ITGM220_JohnDoe_A4datavisualization.zip

4. Submit this file via the Assessment link in the course menu **before** class starts (8:00 p.m. EST/EDT) on due day
5. Post your 1) hand-draw sketch of the project design + 2) final output screenshots / movie clips for peer review to the appropriate module discussion forum by 11:59 p.m. U.S. EST/EDT on due date.

Grading

This Assignment is worth 20 percent of your overall grade in this class. Your Assignment will be graded according to the criteria specified in the Custom Classes rubric, available on the Submissions page.

Concept	Use of Class Hierarchy	Levels of Achievement:	Use of Classes	Code Design, Organization and Comments	Documentation	Functionality, Complexity and Originality.
20	10	10	20	10	10	20

Recommended Project Documentation Structure

HOW	- Document Size: Portrait. (1920 x 1080) or A4 - Presentation Deck: I recommend using presentation tools to create: Keynote, Powerpoint, Google Slides etc. - Export Format: PDF (No Doc / Txt)
WHO	- Reviewer Analysis: Professor will be your primary reviewer. - Presentation: Project will be presented in class via workstations. Students are expected to use a presentation deck to discuss projects.
WHAT	● **TITLE PAGE** (1 page) ○ Student Name, Class Info, Assignment Number ○ Give your work an interesting title. “HOW I LOOK WHEN I SEE MY BEST FRIEND” “ANGRY JIMMY” ● **IDEA / RESEARCH** (1 - 3 pages) ○ Write a statement explaining the idea. ○ Cite influential sources as your inspirations: Painting, photos, movies, poetry etc.

	○ Any relevant pages taken from your sketchbooks. ● **SOLUTION / EVOLUTION** (1 - 3 page) ○ Pencil / Digital Drawing of your intended Results. ○ Post discarded / older ideas as well. ○ Annotate for clarifications. ● **WORK IN PROGRESS** (1 page) ○ Discuss the evolution of the project by showing at least 1 in-progress screenshot. (Just grab the whole screen) ○ You can begin with Pseudocode or a flow chart, but not necessary. ● **FINAL OUTCOME** (1+ page) ○ Record a video clip of the final Output of the code when executed in Processing. ○ Include all unique screenshots to showcase the key feature of your work.. ● **FINAL THOUGHTS** (1+ page) ○ Include a short summary about what you feel you've accomplished in terms of programming ○ What you want to do next with this knowledge.
WHEN	Assignment Submitted to Blackboard before 8:00 pm EST on due day or be considered LATE.

Expectation of an A Assignment:

Before you declare your project finished and ready to deliver, check against the following criterias. Project which qualify for the grade of A should meet most if not all of the following:

Concepts	Written description of concept includes all features to be implemented. At least 5 different visual references are used. Sketches are included and clearly illustrate the desired visual output. All references and sketches are clearly and individually annotated.
Class Hierarchy	At least 2 classes or 2 child classes are defined. Minimal redundancies between the classes. There is no redundancy in the code. The hierarchy pairing solves a major problem in providing functionality of the code.
Use of Classes	Classes are used to solve a major functional limitation that would otherwise affect the readability, reusability or performance of the program.

	Methods and Logics are written with precision.
Code Design Organization	Code is well structured with no redundancy. The code is commented in detail and the comments explain clearly the intent of the code in plain English. The code written goes beyond what is provided in the examples. Code comments are professional. All functions / classes are labeled.
Documentation	Documentations are well organized in a single file. Communicated the development of concept, workflow, code snippets and final output. Materials are organized professionally and are ready to be shared on social media. Ready to be included in the portfolio.
Functionality, Complexity and Originality	The application is interactive and functions as described. All features listed are implemented. The concept is original or. If derivatives from existing work, reflect an original spin of an existing concept. Outcome is outstanding and is ready for review by industry professionals. The code demonstrates a mastery of programming concepts covered in the course. The piece is polished and portfolio ready.