

PROG 101

Introduction to Programming

Model contributed by Janell Baxter

Learn programming concepts as you problem-solve with code, develop algorithms, and create object-oriented applications. Topics covered throughout the semester include data structures, repetition, sequence control structures, program design, testing, and programming best practices.

Class schedule and format

The class will meet “in person” two or three times over the semester.

Course-specific organization, examples and ideas

Asynchronous Online

- Pre-recorded lectures and demonstrations of programming that are approximately 20 minutes in length. The videos have been created by multiple IAM faculty and tutors and provide students with different perspectives on how to program.
 - a. Many students in our program progress through class sessions that are dense in new information/content best when the content is presented in smaller segments followed by a hands-on activity for reinforcement.
 - b. Programming is an art form. There is no “one right way” to solve a programming problem/challenge. Ideally each class there are variations on how to approach and solve the week’s challenge as pre-recorded videos. Students can one watch the approach that best match’s their own mental model.
 - c. Pre-recorded demos allow students more control over the information. They can rewind, fast-forward, pause, or watch again.
- Readings embedded into quizzes. The material is broken up into smaller pieces (versus “wall of text”), and students are asked in real-time to evaluate their comprehension.
- Small assignments ask students to apply the week’s learning and demonstrate their understanding.

Synchronous Online

Weekly synchronous meetings are essential to keep students on track and moving through the course milestones.

- Instructor provides context for the week’s goals.
- Personalized content based on class progression, and student interaction in the asynchronous online material.
- Instructor leads students to break out into small groups (via Microsoft Teams) and manages the de-brief and integration of new ideas. Examples of break out groups:
 - Solving programming challenges using what they have learned from the pre-recorded lectures and demonstrations, and from the assigned readings and other assignments.
 - Playtesting their work in small groups.
 - Discussion groups.

Synchronous In-Person (2 to 3 times)

Students present completed work. This allows everyone to comply with the 6-feet apart guideline, while still preparing students to present their work in front of a live audience.

Example Class Session: Online Synchronous + In-Person Tie-In Later

Class Begins

Students are asked to submit a type of terrain at the beginning of the synchronous class meeting. This provides students with a question: what does terrain have to do with programming? It is also foreshadowing for a collaborative assignment that will take up a large part of the class time.

Brief Lecture

Collection of data stored in arrays and lists, and how that is implemented in C#. Comparison of arrays and lists, and when one might be used over the other.

Assignment Overview

We will create an application that allows a player/user to move about in a world we create. The player will be able to find unique items at each location and meet NPCs.

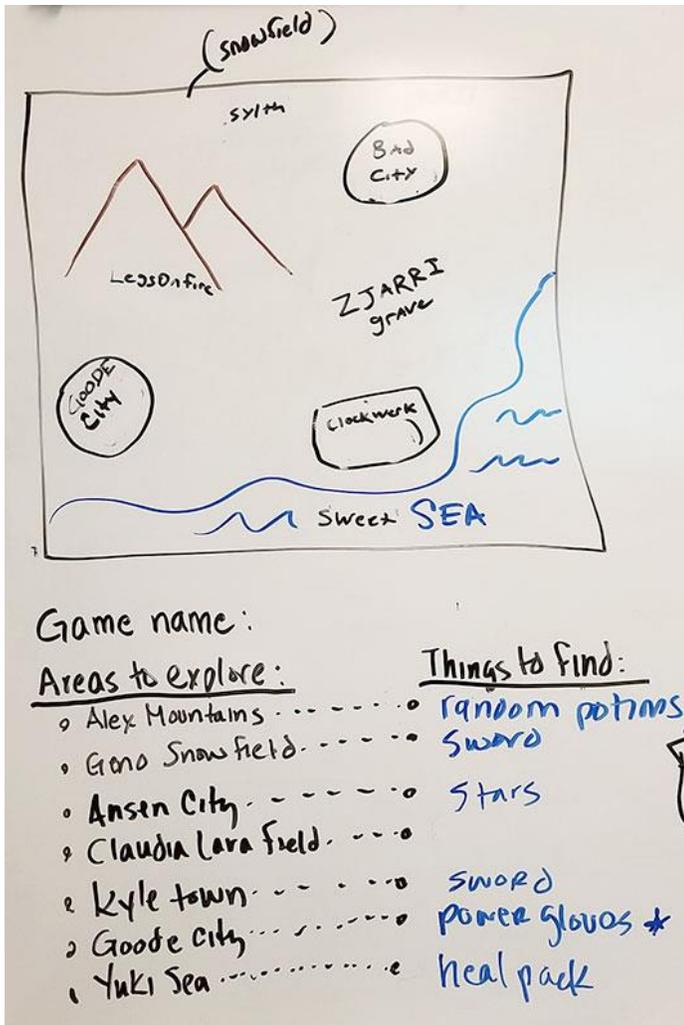
Breakout Groups

20-minute breakout groups: Map, Items, People

Map

Volunteers are requested to create the visual map integrating the terrain types submitted. In-person classes: created on physical whiteboard. Online: Microsoft Office Whiteboard.

Example:



Items

For each location, specify items that the player might find. Identify one or two key items that will serve as gatekeepers for more content (i.e., you can't enter the lava fields until you find the safety suit).

People

Each location should have one or two non-playing characters that the player can meet. Give them names and some sample dialogue.

Collaborative Programming Demo

Instructor builds an example application incorporating the content the students developed in their breakout groups, with input and guidance from the students as the demo progresses (example: what other items should we put in the array for the beach?)

In-Person Synchronous Meeting Tie-In

This assignment helps students learn the skills needed to build a larger adventure game application that they will be presenting in person later in the semester.

The in-person presentation showcases how each student has taken the same fundamental skills and has created their own unique world and player experience.