**CSE 310 – Applied Programming**

**Module Plan**

|  |  |
| --- | --- |
| **Name:** | Benjamin Loughmiller |
| **Date:** | 9/23/2024 |
| **Teacher:** | Nathan Birch |
| **Module # (1-5):** | 1 |

1. Identify which module you have selected to work on. Place an “X” under the “Selected Module” column.

|  |  |
| --- | --- |
| **Modules** | **Selected Module** |
| Cloud Databases |  |
| Data Analysis |  |
| Game Framework |  |
| GIS Mapping |  |
| Mobile App |  |
| Networking |  |
| SQL Relational Databases |  |
| Web Apps |  |
| Language – C++ |  |
| Language – Java |  |
| Language – Kotlin |  |
| Language – R |  |
| Language – Erlang |  |
| Language – JavaScript |  |
| Language – C# | X |
| Language - TypeScript |  |
| Language – Rust |  |
| Choose Your Own Adventure |  |

1. At a high level, describe the software you plan to create that will fulfill the requirements of this module. This may change as you learn more about the technology or language you are learning.
   1. I plan on making a computerized version of the [Tax Collector math game](https://mathforlove.com/wp-content/uploads/2023/02/Beat-the-Tax-Collector.pdf) in the C# language.
   2. In this game, numbers from 1-X (For my program, I plan on making X=50) are listed. The user picks a number and adds it to their score. Any factors of the score are added to the Tax Collector’s score. If a number has no more factors remaining, it is added to the Tax Collector’s score. At the end of the game, if the user has a higher score than the tax collector, they win.
   3. This software will require multiple classes for each part of the program, including taking user input, displaying game information, storing and modifying numbers, and calculating scores.
2. Create a detailed schedule using the table below to complete your selected module during this Sprint. Include details such as what (task), when (time), where (location), and duration. You should also include time to work on your team project. You are expected to spend 16 hours every Sprint working on your individual module, team project, and other activities. Time spent on this individual module should be at least 10 hours.

|  |  |  |
| --- | --- | --- |
|  | **First Week of Sprint** | **Second Week of Sprint** |
| **Monday** | High level planning, Home, 6:00 PM, 30 minutes | Developing, Home, 6:00 PM, 1 hour |
| **Tuesday** | Low level planning, Home, 6:00 PM, 1 hour | Peer review, STC, 11:15 AM, 30 minutes |
| **Wednesday** | Basic development, Home, 6:00 PM, 30 min | New features and bug fixes, Home, 6:00 PM, 1 hour |
| **Thursday** | Menu development, Home, 6:00 PM, 1 hour | New features and bug fixes, Home, 6:00 PM, 2 hours |
| **Friday** | Developing, Home, 6:00 PM, 1 hour | Final peer review, home, 7:00 PM, 30 minutes |
| **Saturday** | Developing, Home, 1:00 PM, 1 hour | Final touches and bug fixes, Home, 1:00 PM, 1 hour |

1. Identify at least two risks that you feel will make it difficult to succeed in this module. Identify an action plan to overcome each of these risks.

Two risks that I feel will make it difficult to succeed in this module are:

1. Storing the numbers and successfully updating it each time the loop is run
   1. To address this risk, I will study ways to store and update the numbers in C#, and find a method that best suits my needs.
2. Displaying the list of numbers to the user in a way that is readable and functional
   1. I will learn different ways of displaying this information to the user, and I will use peer-review to find a view that is satisfactory.