**CSE 310 – Applied Programming**

**Module Plan**

|  |  |
| --- | --- |
| **Name:** | Sam Bingham |
| **Date:** | 9/23/24 |
| **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++ | X |
| Language – Java |  |
| Language – Kotlin |  |
| Language – R |  |
| Language – Erlang |  |
| Language – JavaScript |  |
| Language – C# |  |
| 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.

I plan to make a basic calculator program using C++. The calculator will let the user do simple math operations like adding, subtracting, multiplying, and dividing. The program will ask for two numbers and an operator, then show the result. It will have a simple design with separate functions for each operation, and it will check for things like division by zero and user errors. I will also add extra features, like saving results to a file or handling more complex math.

1. 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** | Set up the C++ environment and start writing the calculator code (6:00 PM - 8:00 PM, Home, 2 hours). | Test and review the entire calculator program (6:00 PM - 7:00 PM, Home, 1 hour). |
| **Tuesday** | Implement basic arithmetic operations for the calculator (6:00 PM - 8:00 PM, Home, 2 hours). | Add optional stretch features like file I/O and advanced operations (6:00 PM - 8:00 PM, Home, 2 hours). |
| **Wednesday** | Test and debug the arithmetic operations (6:00 PM - 7:00 PM, Home, 1 hour). | Continue developing and refining stretch features (6:00 PM - 7:00 PM, Home, 1 hour). |
| **Thursday** | Add input validation and error handling to the calculator (6:00 PM - 8:00 PM, Home, 2 hours). | Debug and finalize any added features for the calculator (6:00 PM - 8:00 PM, Home, 2 hours). |
| **Friday** | Work on the team project (6:00 PM - 8:00 PM, Home, 2 hours). | Work on the team project (6:00 PM - 8:00 PM, Home, 2 hours). |
| **Saturday** | Finalize the basic features of the calculator and ensure functionality (10:00 AM - 12:00 PM, Home, 2 hours). | Final review and testing of the complete calculator project (10:00 AM - 12:00 PM, Home, 2 hours). |

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.

One risk is having trouble with C++ syntax and concepts, like memory management. To fix this, I'll spend extra time studying tutorials and practicing with small examples. I'll also ask for help online or from the class if needed. Another risk is running into tricky bugs. I will fix this by using debugging tools, writing clear code, and taking breaks when I get stuck to avoid burnout.