**CSE 310 – Applied Programming**

**Module Submit**

|  |  |
| --- | --- |
| **Name:** | Michael Johnson |
| **Date:** | 10/19/2024 |
| **Teacher:** | Brother Birch |
| **Module # (1-5):** | #2 |

1. Copy the link to your public GitHub repository here:

https://github.com/Michael0131/CSE310Module2

2. Mark an “X” next to the module you completed:

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

3. Complete the following checklist to make sure you completed all parts of the module. Mark your response with “Yes” or “No”. If the answer is “No” then additionally describe what was preventing you from completing this step.

|  |  |
| --- | --- |
| **Question** | **Your Response** |
| Did you implement the entire set of unique requirements as described in the Module Description document in I-Learn? | Yes |
| Did you write at least 100 lines of code in your software and include useful comments? | Yes |
| Did you use the correct README.md template from the Module Description document in I-Learn? | Yes |
| Did you completely populate the README.md template? | Yes |
| Did you create the video, publish it on YouTube, and reference it in the README.md file? | Yes |
| Did you publish the code with the README.md (in the top-level folder) into a public GitHub repository? | Yes |

4. If you completed a stretch challenge, describe what you completed.

I modified my Weather Forecast App to demonstrate inheritance by using the extends and abstract keywords. I created an abstract class called WeatherService which defines common methods for fetching and processing weather data. The WeatherReport class extends this abstract class and provides specific implementations, allowing for a structured approach to handling weather data while promoting code reusability.

5. How many hours did you spend on this module and the team project this Sprint? Include all time including planning, researching, implementation, troubleshooting, documentation, video production, and publishing.

|  |  |
| --- | --- |
| **Hours spent on this Individual Module** | 16 |
| **Hours spent on your Team Project** | 12 |

6. What learning strategies worked well in this module and what strategies (or lack of strategy) did not work well? How can you improve in the next module?

**Effective Strategies**:

1. **Hands-On Coding**: Actively working on assignments, such as the Weather Forecast App, reinforced my understanding of Java and object-oriented principles.
2. **Incremental Development**: Breaking tasks into smaller parts made it easier to manage and debug code.

**Ineffective Strategies**:

1. **Minimal Documentation**: Relying on memory without taking notes led to confusion with complex topics.
2. **Skipping Foundational Concepts**: This caused gaps in understanding that made advanced topics harder to grasp.

**Improvements for Next Module**:

1. **Consistent Documentation**: I will take detailed notes to enhance retention and understanding.
2. **Structured Study Plan**: Creating a schedule to focus on foundational knowledge before advancing will improve my grasp of concepts.