|  |  |
| --- | --- |
| **Criteria** | **Total Points:**  **50** |
| **Functionality and Implementation** | **Subtotal: 40** |
| The first time the app is run, it automatically creates a database with an empty table. | 3/3 |
| The app loads 3 XML files containing tide data onto your device (These are loaded from the Assets folder) | 3/3 |
| The app stores parsed data from the 3 files in the device’s database | 9/9 |
| Date and time are stored in separate fields | 2/2 |
| User choices are made in the first activity (the one that is started when the app starts) | 2/2 |
| Tide information is displayed in a second activity | 4/4 |
| Users can choose a location and display tide information from that location using data from the database | 7/7 |
| Users can select tide information by date as well as location. | 5/5 |
| The second activity displays complete tide information (date, time, H or L, and height) | 5/5 |

|  |  |
| --- | --- |
| **Code Quality** | **Subtotal: 10** |
| Does the program run without crashing? | 3/3 |
| Is all the game-play logic in a separate class written with good programming practices? | 2/2 |
| Is proper indentation used? | 5/5  (one point deducted for each “no” answer) |
| Are the UI elements named according to convention (example: incrementButton)? |
| Are Java variables, constants, methods, and classes named using descriptive names and correct casing? |
| Have any unnecessary lines of code or files been removed? |
| Are there explanatory comments in the code? |

SCORE: 50/50

Good job! Be careful of having using enter their own date though – very easily breakable.