CS115 Week 6
Testing/Errors/Debugging

BRING A FLASH DRIVE WITH:
Breakout, Trick the Turtle, Die rolling

Possible Errors

- Syntactic Errors – compiler/interpreter finds
- Semantic Errors – logic or formula errors, conversion errors, etc.
- Thematic Errors – Not solving the correct problem!

Testing

- Write test plan BEFORE writing the code using the problem definition and solution design
- A Test Plan specifies the test cases that should be tried, each with input values and the expected output.
- There should be a test case for each condition described or implied in the problem

Test Plan Example

- Write a program to calculate the average of three test scores and display a message indicating whether a student is passing or failing. Failing is an average below 60. If a student is passing with less than a 70 average, a marginal passing message should be displayed.

<table>
<thead>
<tr>
<th>Reason for Test Case</th>
<th>Input Values</th>
<th>Expected Output</th>
<th>Observed Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passing Student</td>
<td>70 80 90</td>
<td>Passing</td>
<td></td>
</tr>
<tr>
<td>Marginal Student</td>
<td>65 75 55</td>
<td>Marginally passing</td>
<td></td>
</tr>
<tr>
<td>Failing Student</td>
<td>45 55 65</td>
<td>Failing</td>
<td></td>
</tr>
<tr>
<td>Invalid Data</td>
<td>-10 30 80</td>
<td>Invalid Input</td>
<td></td>
</tr>
</tbody>
</table>

Debugging (finding and fixing Semantic Errors)

- Breakpoints – Most programming environments allow you to specify a place in the program to pause execution while running.
- Stepping – Most programming environments allow you step through the code line by line, executing each line as you go, maybe even allowing changes to the code as the program is running.
- You can usually examine the current contents of variables to determine if the results of a statement execution are correct.