|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Screenshot | Pass/Fail |
| 1 | Testing Circle class with input radius 1 |  | pass |
| 2 | Testing rectangle class with input width 1, height 1 |  | pass |
| 3 | Testing Square class with input side length 1 |  | pass |
| 4 | Testing triangle class with input side lengths 1,1,1 |  | pass |
| 5 | Testing Cube class with input side length 1 |  | pass |
| 6 | Selecting Cone shape with input height, radius |  | pass |
| 7 | Selecting Cylinder shape with input height and radius |  | pass |
| 8 | Selecting Sphere shape with input radius 1 |  | pass |
| 9 | Selecting Torus shape with input minor radius, major radius |  | pass |
| 10 | Selecting Torus shape with invalid inputs |  | pass |
| 11 | Selecting triangle with invalid inputs |  | pass |
| 12 | Selecting Circle with negative inputs |  | pass |
| 13 | Selecting Cube with negative inputs |  | pass |
| 14 | Selecting square with negative inputs |  | pass |
| 15 | Selecting rectangle with negative inputs |  | pass |
| 16 | Selecting rectangle with different inputs |  | pass |
| 17 | Selecting triangle with different inputs |  | pass |
| 18 | Selecting Circle with non-number inputs |  | pass |

Lesson learned

In this project, I gained valuable insights into object-oriented programming and expanded my knowledge of Java syntax. I now possess a deeper understanding of JavaFX, having created the cone and torus shapes from scratch. One noteworthy syntax feature I learned about is the "double... " which allows a method to access arrays of double type with varying lengths. This feature proved extremely useful for generalizing certain methods, enabling them to work across multiple classes and other calling methods.

Additionally, I acquired skills in constructing a simple GUI application, drawing on the screen, and leveraging classes from previous projects in the current one. Debugging became a significant part of my learning journey, as I encountered compile-time and runtime errors during nearly every project run. I learned how to decipher error messages, identify the root causes, and implement solutions. This iterative development approach compelled me to build each method and class incrementally, rather than writing an entire class or method blindly and hoping for the best.