# Homework: Test Levels and Test Types

## Unit Testing in the Real Life: Testing a Battery

|  |  |
| --- | --- |
| **Test #1** | Take a 1.5v bulb and check if the battery works as expected. The bulb should light up. |
| **Test #2** | Take a multimeter and check and check voltage. It should be ~1.5v. |
| **Test #3** | Take a battery and visually examine it.   * Check its length * Check its diameter * Check if it has a form of cylinder * Check for any leakage, corrosion, damage |
| **Test #4** | Use Universal digital battery test checker to see if the battery works |
| **Test #5** | Try putting the battery in a remote control and see if it works. |
| **Test #6** | Check with a flashlight   * Confirm if the battery size matches the flashlight * Check whether the batteries work as expected (do they light the bulb) |
| **Test #7** | Check the labels on the battery.   * The denoted size should be ”AA” * The denoted voltage should be “1.5V” |
| **Test #8** | Check if the:”+” and “–“ signs are correctly positioned. Use a multimeter. |
| **Test #9** | Environmental tests:   * Low temperature, example 2°C * High temperature, example 40°C |
| **Test #10** | Check the expiration date label. It should be in the future. |
| **Test #11** | Overheating test |

## Unit Testing in the Real Life: Testing a Light Bulb

|  |  |
| --- | --- |
| **Test #1** | We use a battery. The bulb should light up. |
| **Test #2** | Check if the socket is not damaged |
| **Test #3** | Test checking the status of the light bulb: This would check if the light bulb status is updated correctly when it is turned on or off or when the brightness is adjusted. |
| **Test #4** | Test adjusting the brightness of the light bulb: This would check if the light bulb brightness increases and decreases when the appropriate command is given. |

## Unit Testing in the Software World: Age Checker

|  |  |
| --- | --- |
| **Test #1** | AgeChecker(5) 🡪child |
| **Test #2** | AgeChecker(12,9) 🡪 child |
| **Test #3** | AgeChecker(13) 🡪 teenager |
| **Test #4** | AgeChecker(19,5) 🡪 teenager |
| **Test #5** | AgeChecker(20) 🡪 adult |
| **Test #6** | AgeChecker(21) 🡪 adult |
| **Test #7** | AgeChecker(50) 🡪 adult |
| **Test #8** | AgeChecker(64,7) 🡪 adult |
| **Test #9** | AgeChecker(65) 🡪 elder |
| **Test #10** | AgeChecker(75,3) 🡪 elder |
| **Test #11** | AgeChecker(95) 🡪 elder |
| **Test #12** | AgeChecker(150) 🡪 elder |
| **Test #13** | AgeChecker(150,1) 🡪 error |
| **Test #14** | AgeChecker(151) 🡪 error |
| **Test #15** | AgeChecker(-10) 🡪 error |
| **Test #16** | AgeChecker(“Twenty”) 🡪 error / not specified |

## Unit Testing in the Software World: Income Checker

|  |  |
| --- | --- |
| **Test #1** | IncomeChecked(250) 🡪 low |
| **Test #2** | IncomeChecked(1500) 🡪 mid |
| **Test #3** | IncomeChecked(2999) 🡪 mid |
| **Test #4** | IncomeChecked(3000) 🡪 high |
| **Test #5** | IncomeChecked(5000) 🡪 high |
| **Test #6** | IncomeChecked(-200) 🡪 error |

## Integration Testing in the Real Life: Lighting the Bulb

|  |  |
| --- | --- |
| **Test #1** | Implement the following circuit using the provided components.    The bulb should light up. |
| **Test #2** | Implement the following circuit, using the provided components.    Turn on the switch button 🡪 The bulb should light up. |
| **Test #3** | Implement the following circuit, using the provided components.    Turn off the switch button 🡪 The bulb should turn off. |
| **Test #4** | Implement the following circuit with a light switch.    Turn on the switch 🡪 The light bulb should light up. |
| **Test #5** | Implement the following circuit. Try lighting the bulb in a wall power system.    Turn the switch on -🡪 The bulb should light up. |

## \* Integration Testing in the Software World: Ads

|  |  |
| --- | --- |
| **Test #1** | Test the “Login” button while we are on the Home page. We should be led to the “Login form”. |
| **Test #2** | Try to log in with the wrong login credentials. You should remain in the “Login form”. |
| **Test #3** | When we log out we should click the “back” button. We should remain logged out and not being in the account. |
| **Test #4** | Try to access the “Register” link while logged in. You shouldn’t be able to. |
| **Test #5** | Try to access the “Login” link while logged in. You shouldn’t be able to. |
| **Test #6** | Try publishing an ad while not logged in. You should be redirected to the “Login” form. |
| **Test #7** | Try accessing “My ads” while not logged in. You should be redirected to the “Login” form. |
| **Test #8** | Try accessing “Edit Profile” while not logged in. You should be redirected to the “Login” form. |

## \* Integration Testing in the Software World: Credit Risk

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **child** | **teenager** | **adult** | **elder** | **negative** |
| **low** | 100% | 80% | 55% | 60% | error |
| **mid** | 100 | 72% | 37% | 44% | error |
| **high** | 100 | 64% | 19% | 28% | error |
| **negative** | error | error | error | error | error |

Regression test:

CreditRisk(age: 17, income: 0) 🡪 80%

* Bug in the sample calculator

CreditRisk(age: 0, income: 1000) 🡪 100%

## System Testing in the Real Life: Flashlight

|  |  |
| --- | --- |
| **Test #1** | Test the on/off switch. When we flashlight and put new batteries correctly, switch on the light and the flashlight should light up.  Switch off. The bulb should turn off. |
| **Test #2** | Test battery replacement. It should be easy. |
| **Test #3** | Test bulb replacement |
| **Test #4** | Test battery duration. The flashlight should be able to be turned on for at least 1 hour with new batteries. |
| **Test #5** | Test the illumination distance. It should be at least 30m. |
| **Test #6** | Shock resistance test. Try if the flashlight works after falling from the table. |
| **Test #7** | Operation under high/low temperature. |
| **Test #8** | Brightness Test. The light should be bright enough for us to see clearly. |
| **Test #9** | Water Resistance Test. We need to see if we are able to use the flashlight if a few drops of water fall on to it. Or if we use the flashlight with wet hands. |
| **Test #10** | User Interface Test. We should verify that the buttons are responsive and that the modes and settings can be easily switched. |

## System Testing in the Real Life: Digital Scale

|  |  |
| --- | --- |
| **Test #1** | Go on the scale and weigh yourself. |
| **Test #2** | Power on the scale by just stepping on the scale and going away from it. The shown kilos should be 0. |
| **Test #3** | Use a 10kg water bottle to weigh it. It should show 10kg. |
| **Test #4** | Verify that the weight is displayed clearly and accurately on the scale's screen under different lighting conditions. |
| **Test #5** | Try weighting yourself around 5 times sequentially to ensure that if many people want to use the scale one after another, they would be able to. |
| **Test #6** | Test the scale's durability by dropping it from a specified height and ensure that it continues to function as expected. |
| **Test #7** | If there are other units of measurement (grams, ounces, pounds) try switching between them and perform the tests again to make sure everything works as expected. Test switching between different units of measurement and ensure that the weight displayed on the scale changes accordingly. |

## System Testing in the Software World: Number Calculator

|  |  |
| --- | --- |
| **Test #1** | Calc(5, +, 3) 🡪 8 |
| **Test #2** | Calc(5, -, 3) 🡪 2 |
| **Test #3** | Calc(5, +, 0) 🡪 5 |
| **Test #4** | Calc(Infinity, +, 1) 🡪 Infinity |
| **Test #5** | Calc(-Infinity, +, 1) 🡪 Infinity |
| **Test #6** | Calc(Infinity, +, Infinity ) 🡪 Infinity |
| **Test #7** | Calc(Text, +, 1) 🡪 Invalid input |
| **Test #8** | Calc(-Infinity, +,- Infinity ) 🡪 -Infinity |
| **Test #9** | Calc(1000000000000000, +,5) 🡪 1000000000000000 Failed |
| **Test #10** | Calc(70, +, 5,5) 🡪 75,5 |
| **Test #10** | Calc(2, \*, 2) 🡪 4 |
| **Test #10** | Calc(-Infinity, +, -1) 🡪 -Infinity |
| **Test #11** | Calc(10text, +, 1) 🡪 11  *\*Tip: It may show “Invalid input”* |
| **Test #12** | Calc(10, /, 0) 🡪 Infinity Failed |
| **Test #13** | Calc(0, /, 0) 🡪 Invalid calculation.  \*Tip: It should be ***indeterminate*** instead |
| **Test #14** | Calc(1, -select an operation, 1) 🡪 Invalid operation.  *\*Tip: We may add “Please choose an operation” message.* |
| **Test #15** | Calc(-1, /, 0) 🡪 Infinity |

## Acceptance Testing in the Real Life: Flashlight

|  |  |
| --- | --- |
| **Test #1** | The customer takes the flashlight, switches it on and off to make sure it works as expected. |
| **Test #2** | The customer checks the flash illumination. |
| **Test #3** | The customer checks how easy it is to change the batteries. |
| **Test #4** | The customer checks the brightness of the flashlight. |
| **Test #5** | The customer checks if the flashlight is not too heavy to use. |
| **Test #6** | The customer checks if the flashlight is in a good condition and appearance. |

## Acceptance Testing in the Real Life: Digital Scale

|  |  |
| --- | --- |
| **Test #1** | The customer should go on the scale and weigh themselves. |
| **Test #2** | The customer can use a 10liter water bottle to make sure it shows 10kg. |
| **Test #3** | The customer can try weighing themselves a few times, at least 5, to make sure the scale would work if many people want to use it one after another. |
| **Test #4** | If there are other units of measurement, the customer should try switching between them and weighting themselves again. |

## Acceptance Testing in the Software World: Number Calculator

|  |  |
| --- | --- |
| **Test #1** | Try with correct statements.   * 2+2=4 * 5-3=2 * 10/2=5 * 10\*10=100 |
| **Test #2** | Try dividing by 0: 130/0=Infinity |
| **Test #3** | Try an obvious incorrect statement: text+assd=Invalid input |

## Functional and Non-Functional Tests: Flashlight

|  |  |
| --- | --- |
| **Functional Tests** | **Non-Functional Tests** |
| Switching on/off the light | Battery duration |
| Battery replacement | Illumination distance |
| Switching modes buttons | Shock resistance |
|  | High/low temperature |
|  | Water resistant |