**Individual Assignment - Lesson 4**

Aaron Keller

CIT260 05

**Step 1:**

**Task:** calculate the damage inflicted in a battle

**Input:** attack rating, defense rating

**Output:** The amount of damage dealt in health points

**Validation:** The amount of damage points dealt must be zero or a positive number. Return a value of 0 if value is outside of requirements.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **calcDamage Test Matrix** | | | | | | | |
|  | **Test Cases** | | | | | | |
|  | **Valid Input** | **Invalid Input** | | | **Boundaries** | | |
|  | **1** | **2** | **3** | **4** | **5** | **6** | |
| **Input** |  | | | | | | |
| attackRating | 18 | 14 | -1 | 15 | 0 | | 25 |
| defenseRating | 13 | 14 | 13 | 16 | 25 | | 0 |
| **Output** |  | | | | | | |
| damagePoints | 5 | 0 | 0 | 0 | 0 | | 25 |
| **Error** |  | attackRating | attackRating | attackRating | 0..25 | | 0..25 |
|  |  | value is equal to defenseRating value | value is less than 0 | value is less than defenseRating value |  | |  |

**Step 2:**

calcDamage (attackRating, defenseRating): int

BEGIN

IF (attackRating < 0 OR attackRating <= defenseRating) THEN

RETURN 0

damagePoints = attackRating – defenseRating

RETURN damagePoints

END

**Step 3:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **calcDamage Test Memory Test** | | | | | | | | |
|  | **Test Cases** | | | | | | | |
|  | **1** | **2** | **3** | **4** | **5** | | **6** | |
| **Input** |  | | | | | | | |
| attackRating | 18 | 14 | -1 | 15 | | 0 | | 25 |
| defenseRating | 13 | 14 | 13 | 16 | | 25 | | 0 |
| **Output** |  | | | | | | | |
| damagePoints | 5 | 0 | 0 | 0 | | 0 | | 25 |

**Step 4:**

**Task:** calculate the new number of health points

**Input:** healthPoints, damagePoints

**Output:** newHealthPoints

**Validation:** The amount of health points dealt must be zero or a positive number. The new number of health points must not be higher than the original number of health points. Return error of -1 if value is outside of requirements.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **calcNewHealth Test Matrix** | | | | | | | | |
|  | **Test Cases** | | | | | | | |
|  | **Valid Input** | **Invalid Input** | | | **Boundaries** | | | |
|  | **1** | **2** | **3** | **4** | **5** | | **6** | |
| **Input** |  | | | | | | | |
| healthPoints | 18 | 14 | -1 | 15 | | 0 | | 25 |
| damagePoints | 13 | 14 | 13 | 16 | | 25 | | 0 |
| **Output** |  | | | | | | | |
| newHealthPoints | 5 | 0 | 0 | 0 | | 0 | | 25 |
| **Error** |  | attackRating | attackRating | attackRating | | 0..25 | | 0..25 |
|  |  | value is equal to defenseRating value | value is less than 0 | value is less than defenseRating value | |  | |  |

**Step 5:**

calcNewHealth (healthPoints, damagePoints): int

BEGIN

IF (healthPoints < damagePoints) THEN

RETURN -1

IF (healthPoints = damagePoints) THEN

RETURN 0

newHealthPoints = healthPoints – damagePoints

RETURN newHealthPoints

END

**Step 6:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **calcNewHealth Test Memory Test** | | | | | | | | |
|  | **Test Cases** | | | | | | | |
|  | **1** | **2** | **3** | **4** | **5** | | **6** | |
| **Input** |  | | | | | | | |
| healthPoints | 18 | 14 | -1 | 15 | | 0 | | 25 |
| damagePoints | 13 | 14 | 13 | 16 | | 25 | | 0 |
| **Output** |  | | | | | | | |
| newHealthPoints | 5 | 0 | -1 | -1 | | 0 | | 25 |

**Step A:**

**Task:** calculate the adjusted attackRating after modifications

**Input:** healthPoints, weaponEquipped, experiencePoints, attackRating

**Output:** modAttackRating

**Validation:** The modAttackRating must be zero or a positive number. Return error of -1 if value is outside of requirements.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **calcAttack Test Matrix** | | | | | | | | |
|  | **Test Cases** | | | | | | | |
|  | **Valid Input** | **Invalid Input** | | | **Boundaries** | | | |
|  | **1** | **2** | **3** | **4** | **5** | | **6** | |
| **Input** |  | | | | | | | |
| healthPoints | 18 | 14 | -1 | 15 | | 0 | | 25 |
| attackRating |  |  |  |  | |  | |  |
| weaponDamage |  |  |  |  | |  | |  |
| experiencePoints |  |  |  |  | |  | |  |
| **Output** |  | | | | | | | |
| modAttackRating |  |  |  |  | |  | |  |
| **Error** |  | attackRating | attackRating | attackRating | | 0..25 | | 0..25 |
|  |  | value is equal to defenseRating value | value is less than 0 | value is less than defenseRating value | |  | |  |

**Step B:**

calcAttack (healthPoints, attackRating, weaponEquipped, experiencePoints): int

BEGIN

IF (healthPoints < damagePoints) THEN

RETURN -1

IF (healthPoints = damagePoints) THEN

RETURN 0

newHealthPoints = healthPoints – damagePoints

RETURN newHealthPoints

END

**Step C:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **calcNewHealth Test Memory Test** | | | | | | | | |
|  | **Test Cases** | | | | | | | |
|  | **1** | **2** | **3** | **4** | **5** | | **6** | |
| **Input** |  | | | | | | | |
| healthPoints | 18 | 14 | -1 | 15 | | 0 | | 25 |
| damagePoints | 13 | 14 | 13 | 16 | | 25 | | 0 |
| **Output** |  | | | | | | | |
| newHealthPoints | 5 | 0 | -1 | -1 | | 0 | | 25 |