**Dr. Dan Bennett**

**Class: StackT**

**Method Prototype: StackT()**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Testing for** | **Test Case** | **Input/Test value** | **Expected Outcome** | **Observed Result** |
| ST1 | Initializing the stack | StackT s | N/A | The stack is initialized with a size of 0 | As expected |

**Method Prototype: void Push(ItemT i)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Testing for** | **Test Case** | **Input/Test value** | **Expected Outcome** | **Observed Result** |
| P1 | Putting ItemTs on the stack | RationalT a(i) | a = 0 | 0 | As expected |
| P2 |  | RationalT a(i) | a = 1 | 1 | As expected |
| P3 |  | RationalT a(i) | a = 2 | 2 | As expected |
| P4 |  | RationalT a(i) | a = 3 | 3 | As expected |
| P5 |  | RationalT a(i) | a = 4 | 4 | As expected |
| P6 |  | RationalT a(i) | a = 5 | 5 | As expected |

**Method Prototype: ItemT Pop(void)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Testing for** | **Test Case** | **Input/Test value** | **Expected Outcome** | **Observed Result** |
| PP1 | Removing ItemTs from the stack | RationalT a(i) | a = 9 | 9 | As expected |
| PP2 |  | RationalT a(i) | a = 8 | 8 | As expected |
| PP3 |  | RationalT a(i) | a = 7 | 7 | As expected |
| PP4 |  | RationalT a(i) | a = 6 | 6 | As expected |
| PP5 |  | RationalT a(i) | a = 5 | 5 | As expected |
| PP6 |  | RationalT a(i) | a = 4 | 4 | As expected |

**Method Prototype: ItemT Top(void) const**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Testing for** | **Test Case** | **Input/Test value** | **Expected Outcome** | **Observed Result** |
| T1 | Outputting the top value on the stack | s.Top() | s.Size() = 1; s[0] = 0 | 0 | As expected |
| T2 |  | s.Top() | s.Size() = 2; s[1] = 1 | 1 | As expected |
| T3 |  | s.Top() | s.Size() = 3; s[2] = 2 | 2 | As expected |
| T4 |  | s.Top() | s.Size() = 4; s[3] = 3 | 3 | As expected |
| T5 |  | s.Top() | s.Size() = 5; s[4] = 4 | 4 | As expected |
| T6 |  | s.Top() | s.Size() = 6; s[5] = 5 | 5 | As expected |

**Method Prototype: int Size(void) const**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Testing for** | **Test Case** | **Input/Test value** | **Expected Outcome** | **Observed Result** |
| S1 | Outputting the size of the stack | s.Size() | s.Size() = 1 | S’s size is: 1 | As expected |
| S2 |  | s.Size() | s.Size() = 2 | S’s size is: 2 | As expected |
| S3 |  | s.Size() | s.Size() = 3 | S’s size is: 3 | As expected |
| S4 |  | s.Size() | s.Size() = 4 | S’s size is: 4 | As expected |
| S5 |  | s.Size() | s.Size() = 5 | S’s size is: 5 | As expected |
| S6 |  | s.Size() | s.Size() = 6 | S’s size is: 6 | As expected |

**Method Prototype: bool IsEmpty(void) const**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Testing for** | **Test Case** | **Input/Test value** | **Expected Outcome** | **Observed Result** |
| IE1 | Determining if the stack is empty | s.IsEmpty() | s.Size() > 0 | Item removed from stack and “S’s size is: “ s.Size() | As expected |
| IE2 |  | s.IsEmpty() | s.Size() = 0 | Stack is empty | As expected |

**Method Prototype: bool IsFull(void) const**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key** | **Testing for** | **Test Case** | **Input/Test value** | **Expected Outcome** | **Observed Result** |
| IF1 | Determining if the stack is full | s.IsFull() | s.Size() < STACK\_MAX | Item at top of stack and “S’s size is: “ s.Size() | As expected |
| IF2 |  | s.IsFull() | s.Size() = STACK\_MAX | Stack is full | As expected |