**DATA STRUCTURE TEST**

Problem Statement : Bob-The dishwasher

**import** java.util.NoSuchElementException;

**public** **class** WashTheDishes {

**protected** **int** arr[];

**protected** **int** top, size, len;

/\* Constructor for WashTheDishes \*/

**public** WashTheDishes(**int** n)

{

size = n;

len = 0;

arr = **new** **int**[size];

top = -1;

}

/\* Function to check if stack is empty \*/

**public** **boolean** isEmpty()

{

**return** top == -1;

}

/\* Function to check if stack is full \*/

**public** **boolean** isFull()

{

**return** top == size -1 ;

}

/\* Function to get the size of the stack \*/

**public** **int** getSize()

{

**return** len ;

}

/\* Function to check the top element of the stack \*/

**public** **int** peek()

{

**if**( isEmpty() )

**throw** **new** NoSuchElementException("There are no dish present on the shelf");

**return** arr[top];

}

/\* Function to add an element to the stack \*/

**public** **void** push(**int** i)

{

**if**(top + 1 >= size)

**throw** **new** IndexOutOfBoundsException("There is no extra place to put more dishes on the shelf");

**if**(top + 1 < size )

arr[++top] = i;

len++ ;

}

/\* Function to delete an element from the stack \*/

**public** **int** pop()

{

**if**( isEmpty() )

**throw** **new** NoSuchElementException("There are no dish present on the shelf");

len-- ;

**return** arr[top--];

}

/\* Function to display the status of the stack \*/

**public** **void** display()

{

System.***out***.print("\nStack = ");

**if** (len == 0)

{

System.***out***.print("Empty\n");

**return** ;

}

**for** (**int** i = top; i >= 0; i--)

System.***out***.print(arr[i]+" ");

System.***out***.println();

}

}

**import** java.util.Scanner;

**public** **class** CleanDishes {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner scan = **new** Scanner(System.***in***);

System.***out***.println("Data Structure Test\n");

System.***out***.println("Enter the number of dishes to wash ");

**int** n = scan.nextInt();

WashTheDishes stk = **new** WashTheDishes(n);

**char** ch;

**do**{

System.***out***.println("\nOperations");

System.***out***.println("1.Please place washed dish onto the shelf:");

System.***out***.println("2.Dish Withdrawn from the shelf");

System.***out***.println("3.Dish number of the top dish present on the shelf");

System.***out***.println("4.Check whether shelf is empty?");

System.***out***.println("5.Check whether shelf is full?");

System.***out***.println("6.No of dishes on the shelf : ");

**int** choice = scan.nextInt();

**switch** (choice)

{

**case** 1 :

System.***out***.println("Enter the total capacity of shelf");

**try**

{

stk.push( scan.nextInt() );

}

**catch** (Exception e)

{

System.***out***.println("Error : " + e.getMessage());

}

**break**;

**case** 2 :

**try**

{

System.***out***.println("Dish withdrawn = " + stk.pop());

}

**catch** (Exception e)

{

System.***out***.println("Error : " + e.getMessage());

}

**break**;

**case** 3 :

**try**

{

System.***out***.println("Dish present on the top of the shelf = " + stk.peek());

}

**catch** (Exception e)

{

System.***out***.println("Error : " + e.getMessage());

}

**break**;

**case** 4 :

System.***out***.println("No dishes on the shelf = " + stk.isEmpty());

**break**;

**case** 5 :

System.***out***.println("No more place remaining to place the dish on the shelf = " + stk.isFull());

**break**;

**case** 6 :

System.***out***.println("Dishes on the shelf = " + stk.getSize());

**break**;

**default** :

System.***out***.println("Wrong Entry \n ");

**break**;

}

/\* display stack \*/

stk.display();

System.***out***.println("\nDo you want to continue (Type y or n) \n");

ch = scan.next().charAt(0);

} **while** (ch == 'Y'|| ch == 'y');

}

}