## 

## **Computing 2: 2021-2022**

## DSA I- Project Assignment

Table of contents

* Description of the problem .3
* Detailed description of underlying .4

data structures used

* Pseudocode .5
* Extra operations .7
* Java code .8
* All methods/routines Used .17
* Test data/Sample execution screenshots .18
* Conclusion .23

Description of the problem

A Set is a container capable of holding a number of entries. Each entry is a key and keys must be distinct. It supports the following operations.

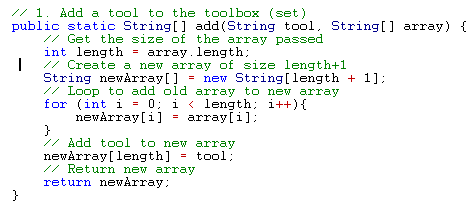
* **add(k,S):** Inserts a new entry with key k into the set S and return the new entry. It must check that item not already in set as duplicates not permitted.
* **contains(k, S):** Return true if this set S contains the specified element k i.e. if k is a member of this set.
* **removeAll(remSet,S)** : Remove from this set S all of its elements that are contained in the specified set remSet. This operation effectively modifies this set so that its new value is the set difference of the two sets.
* **print(S)** : print the contents of set S

All operations are to be options on a menu – provide clear instructions on User Interface as to how to use it.

Detailed description of underlying

Data structures used

A container object that stores a fixed number of values of a single type is called an array. The length of an array is determined at the time of its creation. Its length is fixed after it is created. An element is the name given to each item in an array, and each element is accessed by its integer index.



This is an example of how to add things to a given array

Pseudocode

add(k,s)

add(String tool, String[] array) {

length = array.length

int newArray[] = new String[length + 1];

for (int i = 0; i < length; i++){

newArray[i] = array[i];

}

newArray[length] = tool;

return newArray;

}

contains(k,s) contains(String tool, String[] array)

for(int i = 0; i < array.length; i++){

if (tool = array)){

return true;

else

return false;

Remove()

InputNextLine();

print("Enter a tool: ");

tool = InputNextLine();

if(!contains(tool, toolbox)){

print(tool + " is NOT in toolbox");

}

else

{

toolbox = removeAll(tool, toolbox);

Print()

print(String[] array) {

for(int i = 0; i < array.length; i++){

print(array[i] + " ");

Extra Operations Used

update ()

for(int i = 0; i < array.length; i++){

if(oldTool.equals(array[i])){

array[i] = newTool;

}

}

return array;

}



Java Code Used

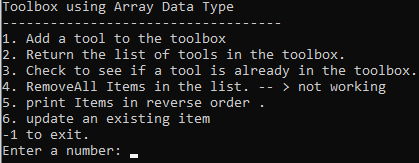
|  |
| --- |
| import java.util.Scanner;  public class Main {  // 1. Add a tool to the toolbox (set)  public static String[] add(String tool, String[] array) {  // Get the size of the array passed  int length = array.length;  // Create a new array of size length+1  String newArray[] = new String[length + 1];  // Loop to add old array to new array  for (int i = 0; i < length; i++){  newArray[i] = array[i];  }  // Add tool to new array  newArray[length] = tool;  // Return new array  return newArray;  }  //-----------------------------------------------------------------------------------------------------------  // 2. Print out elements in toolbox  public static void print(String[] array) {  for(int i = 0; i < array.length; i++){  System.out.print(array[i] + " ");  }  }  //-----------------------------------------------------------------------------------------------------------  // 3. Check to see if tool is in toolbox  public static boolean contains(String tool, String[] array) {  for(int i = 0; i < array.length; i++){  if(tool.equals(array[i])){  return true;  }  }  return false;  }  //-----------------------------------------------------------------------------------------------------------  // 4. RemoveAll items in the tool box  // Not Working  //------------------------------------------------------------------------------------------  // 5. Print List in reverse order  //code below  //-----------------------------------------------------------------------------------------------------------  // 6. Update existing tool  public static String[] update(String newTool, String oldTool, String[] array) {  for(int i = 0; i < array.length; i++){  if(oldTool.equals(array[i])){  array[i] = newTool;  }  }  return array;  }    //-----------------------------------------------------------------------------------------------------------      // print the menu  public static void printMenu() {  System.out.println("Toolbox using Array Data Type");  System.out.println("-----------------------------------");  System.out.println("1. Add a tool to the toolbox");  System.out.println("2. Return the list of tools in the toolbox.");  System.out.println("3. Check to see if a tool is already in the toolbox.");  System.out.println("4. RemoveAll Items in the list. -- > not working ");  System.out.println("5. print Items in reverse order .");  System.out.println("6. update an existing item ");  System.out.println("-1 to exit.");  System.out.print("Enter a number: ");  }  //----------------------------------------------------------------------------------  public static void main(String[] args) {  String[] toolbox = new String[0];  Scanner sc = new Scanner(System.in);  printMenu();  boolean menu = true;  while (menu) {  int choice = sc.nextInt();    //-----------------------------------------------------------------------------------------------------------  if (choice == 1)    {  // 1. Add a tool to toolbox  sc.nextLine();  System.out.print("Enter a tool: ");  String tool = sc.nextLine();  if(contains(tool, toolbox)){  System.out.println(tool + " is already in toolbox");  }  else  {  toolbox = add(tool, toolbox);  }    System.out.print("Press 0 for menu, -1 to exit: ");  }  //------------------------------------------------------------------------------------------------------------      if (choice == 2) {  // 2. Print out the toolbox  print(toolbox);  System.out.println();  System.out.print("Press 0 for menu, -1 to exit: ");  }  //------------------------------------------------------------------------------------------ if (choice == 3)  {  // 3. Check to see if tool is in toolbox  sc.nextLine();  System.out.print("Enter a tool: ");  String tool = sc.nextLine();  if(contains(tool, toolbox))  {  System.out.println(tool + " is in toolbox");  }  else {  System.out.println(tool + " is NOT in toolbox");  }  System.out.print("Press 0 for menu, -1 to exit: ");  }  //------------------------------------------------------------------------------------------  /\* if (choice == 4) {  // 4. Remove a tool to toolbox  sc.nextLine();  System.out.print("Enter a tool: ");  String tool = sc.nextLine();  if(!contains(tool, toolbox)){  System.out.println(tool + " is NOT in toolbox");  } else{  toolbox = removeAll(tool, toolbox);  }  System.out.print("Press 0 for menu, -1 to exit: ");\*/  //-----------------------------------------------------------------------------------------------------------  if (choice == 5) {  // 5. print in reverse order    for (int i = toolbox.length-1; i >= 0; i--) {  System.out.print(toolbox[i] + " ");  }  System.out.print("Press 0 for menu, -1 to exit: ");  }  //-----------------------------------------------------------------------------------------------------------  **if (choice == 6) {**  **// 6. Remove a tool to update**  **sc.nextLine();**  **System.out.print("Enter a tool to update: ");**  **String oldTool = sc.nextLine();**  **if(!contains(oldTool, toolbox)){**  **System.out.println(oldTool + " is NOT in toolbox");**  **}**  **else**  **{**    **System.out.print("Enter a new tool: ");**  **String newTool = sc.nextLine();**  **if(contains(newTool, toolbox)){**  **System.out.println(newTool + " is already in toolbox");**  **}**    **else**    **{**  **toolbox = update(newTool, oldTool, toolbox);**  **}**  **}**    **System.out.print("Press 0 for menu, -1 to exit: ");**  **}**  //-------------------------------------------------------------------------------------------------------------    if (choice == 0) {  printMenu();  }    if(choice == -1){  menu = false;  }  }  }  } |

All methods/routines Used

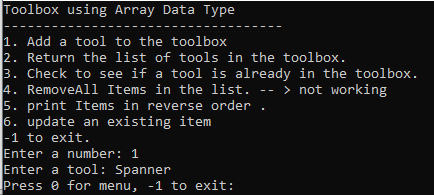
* **add(k,S):** Inserts a new entry with key k into the set S and return the new entry. It must check that item not already in set as duplicates not permitted.
* **contains(k, S):** Return true if this set S contains the specified element k i.e. if k is a member of this set.
* **removeAll(remSet,S)** : Remove from this set S all of its elements that are contained in the specified set remSet. This operation effectively modifies this set so that its new value is the set difference of the two sets.
* **print(S)** : print the contents of set S.
* **reverseOrder(S)** : Takes the inputs of set S and put’s it in reverse order.
* **Update(S) :** The user is asked to Select an already existing item in the set and then given the opportunity to change set item.

Test Data/Sample Data Screenshots

Menu

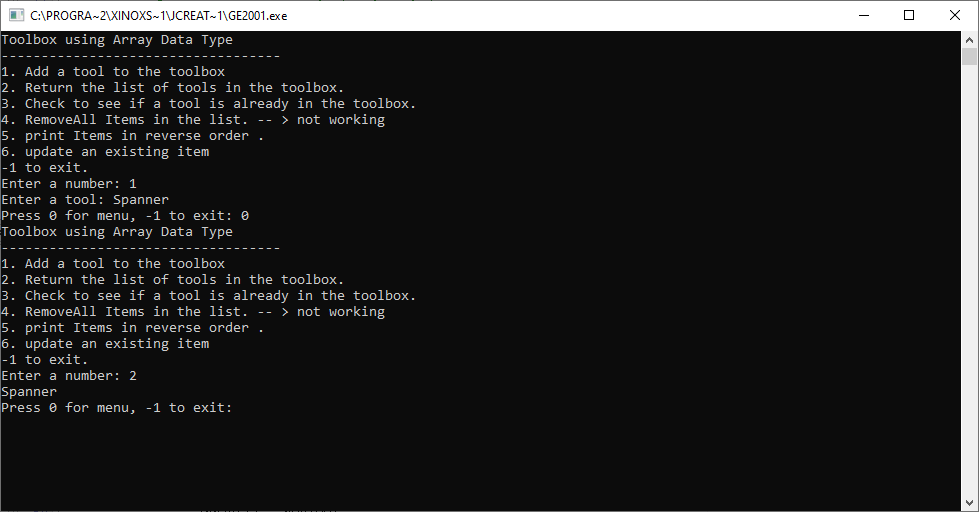


Output 1

 Add a new item

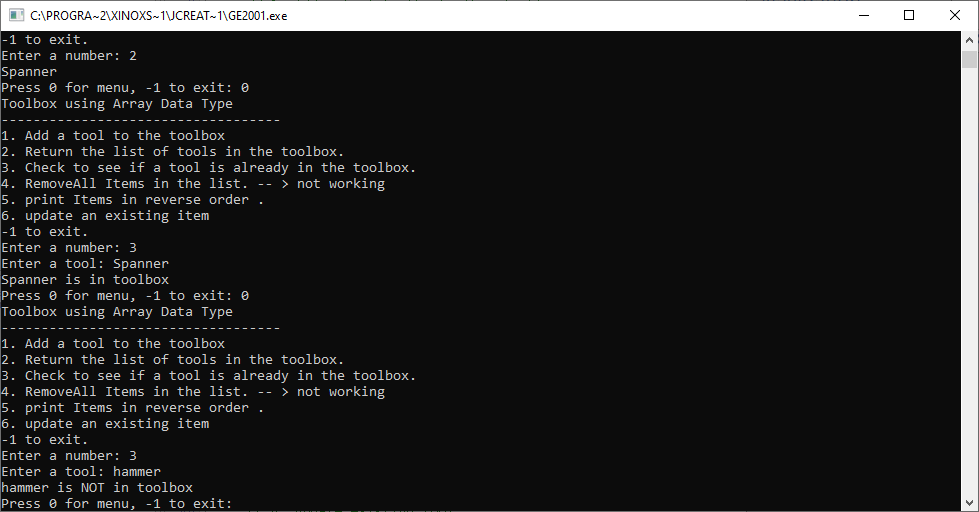
Output 2

Check to see is the item in the list



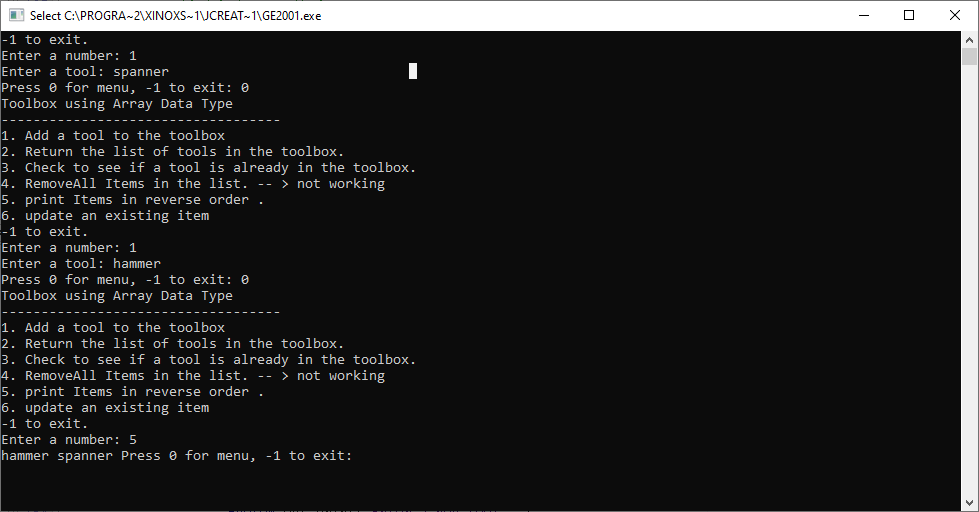
Output 3

Check to see if the tool already exists



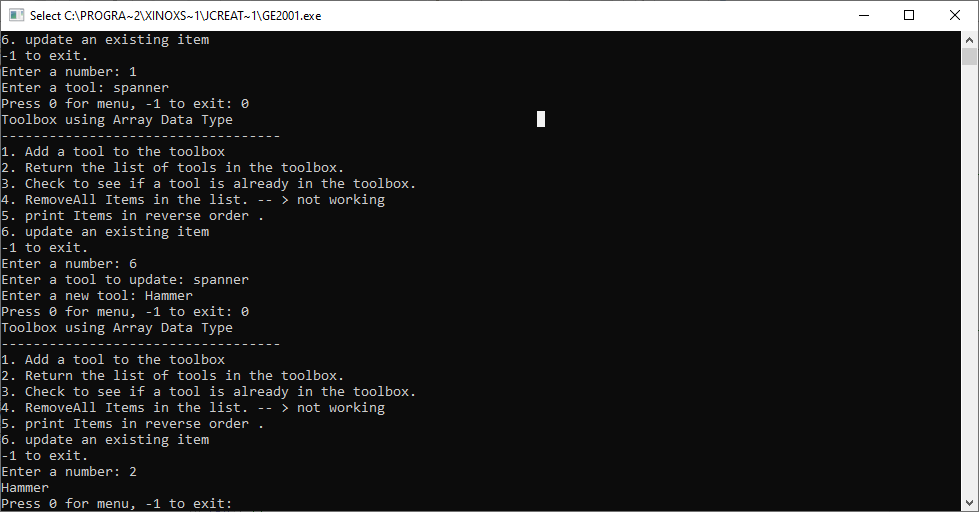
Output 5

Display tools in reverse order from input



Output 6

Edit an already existing tool



Conclusion

**The menu driven application to implement a toolbox for the SET data type and its operations allows a user to use the following operations:**

* **add(k,S):** Inserts a new entry with key k into the set S and return the new entry. It must check that item not already in set as duplicates not permitted.
* **contains(k, S):** Return true if this set S contains the specified element k i.e. if k is a member of this set.
* **removeAll(remSet,S)** : Remove from this set S all of its elements that are contained in the specified set remSet. This operation effectively modifies this set so that its new value is the set difference of the two sets.
* **print(S)** : print the contents of set S
* **reverseOrder(s)** : Takes the input of the user and put’s the data in reverse order.
* **Update(S) :** The user is asked to Select an already existing item in the set and then given the opportunity to change set item.