**CSC 3020 – Java Programming**

**Homework 4 – Mike Torchia**

**25 points – Due February 22, 10am**

**Late deadline is February 24, 11:59pm, but 20% off**

**a)** Save this document with your name and the homework number somewhere in the file name.

**b)** Type/paste your answers into the document.

**c)** Submit this document to the Blackboard item where you downloaded this document.

**1) [15 points]** You've been hired by *City Skies* to write a Java console application that manages city names. Use an array of size 30 to store the names. Present the following menu to the user:

City Skies Menu

1 – Add city

2 – Change city name

3 – Delete city

4 – List cities

9 – Exit

Enter an option:

Continue to read an option, process it, and display the menu until the user enters option 9. Here are option descriptions:

● Add city – prompt for a city name. If the city is already in the list, don’t add it and tell the user. If the city is not in the list, add it and tell the user. You don’t need to test if the array is full before adding a city.

● Change city name – prompt for a city (array) index. If the index is invalid, tell the user. If the index is valid, prompt for the new city name. If the new city name is already in the list, don’t add it and tell the user. If the new city name is not in the list, add it and tell the user.

● Delete city – prompt for a city (array) index. If the index is invalid, tell the user. If the index is valid, delete the city and move all the cities after the index up one to close the gap.

● List cities – use printf to format the list in two columns with the first column containing an array index and the second column containing the city name. Also, show a city count after the list.

Redisplay the menu after each option is processed. Start the array with three values already in it: Detroit, Dearborn, and Windsor. Use these inputs for your last run:

Option Value(s)

1 Ann Arbor

4

2 6

2 3, Dexter

4

2 3, Dearborn

3 6

3 0

4

*[your program code here]\**

**package** homework\_4;

**import** java.util.Scanner;

**public** **class** Homework4 {

**static** Scanner *keyboard*=**new** Scanner(System.***in***);

**public** **static** **int** printMenu() //function for printing menu and returning chioce

{

String choice;

**int** \_choice;

System.***out***.println("City Skies Menu");

System.***out***.println("1 - Add City");

System.***out***.println("2 - Change City Name");

System.***out***.println("3 - Delete City");

System.***out***.println("4 - List Cities");

System.***out***.println("9 - Exit");

System.***out***.println("Enter An Option: ");

choice=*keyboard*.nextLine();

\_choice=Integer.*parseInt*(choice);

**return** \_choice;

}

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

String[] cities=**new** String[30]; //Declare Variables

**for**(**int** i=0;i<cities.length;i++) //Populate the array with empty values

{

cities[i]="";

}

cities[0]="Detroit";

cities[1]="Dearborn";

cities[2]="Windsor";

**int** choice=0;

String tempCity;

String \_tempIndex;

**int** tempIndex;

**int** cityCount=3;

System.***out***.println("Question 1");

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

**while**(choice!=9) //Keeps the program Running until user enters 9

{

choice=*printMenu*(); //see printMenu function above

**if**(choice==1)

{

//Add City

System.***out***.println("Enter the city you'd like to add: ");

tempCity=*keyboard*.nextLine();

**for**(**int** i=0;i<cities.length;i++) //loop until empty index is found. add city

{

**if**(cities[i].isEmpty())

{

cities[i]=tempCity;

i=cities.length;

}

}

cityCount++;

}

**else** **if**(choice==2)

{

//Change City Name

Boolean cityExist=**false**; //if the city exists flag

System.***out***.println("Which city index would you like to change: ");

\_tempIndex=*keyboard*.nextLine();

tempIndex=Integer.*parseInt*(\_tempIndex);

**while**(tempIndex>30 || tempIndex<0) //validation loop

{

System.***out***.println("The value you entered was out of range. Try again");

\_tempIndex=*keyboard*.nextLine();

tempIndex=Integer.*parseInt*(\_tempIndex);

}

**if**(cities[tempIndex].isEmpty())

{

System.***out***.println("Index you are trying to edit is invalid/empty");

cityExist=**true**; //cities doesnt exisit but set this to true to control the output

**continue**;

}

System.***out***.println("What would you like to change the name to: ");

tempCity=*keyboard*.nextLine();

**for**(**int** i=0;i<cities.length;i++) //loop through and see if the city exists. If so tell user and change flag

{

**if**(cities[i].equals(tempCity))

{

System.***out***.println("That city is already accounted for");

cityExist=**true**;

}

}

**if**(cityExist==**false**) //if city doesnt exist. edit the index to what the user entered

{

cities[tempIndex]=tempCity;

cityExist=**false**;

}

}

**else** **if**(choice==3)

{

//Delete City

**boolean** emptyIndex=**false**;

System.***out***.println("Which city index would you like to delete: ");

\_tempIndex=*keyboard*.nextLine();

tempIndex=Integer.*parseInt*(\_tempIndex);

**while**(tempIndex>30 || tempIndex<0) //validation loop

{

System.***out***.println("The value you entered was out of range. Try again");

\_tempIndex=*keyboard*.nextLine();

tempIndex=Integer.*parseInt*(\_tempIndex);

}

**if**(cities[tempIndex].isEmpty())

{

System.***out***.println("There is no city to delete. Index is empty");

emptyIndex=**true**;

}

**if**(emptyIndex==**false**)

{

cities[tempIndex]=""; //delete value of tempIndex

**for**(**int** i=tempIndex;i<cities.length;i++)

{

**if**(i==cities.length-1) //If the for loop is at the last value of the array

{ //do the swap like this to avoid out of range errors

cities[i-1]=cities[i];

cities[i]="";

}

**else**

{

cities[i]=cities[i+1];

}

}

cityCount--;

}

}

**else** **if**(choice==4)

{

//List Cities

System.***out***.printf("%-10s %-13s %n","Index","City Name");

**for**(**int** i=0;i<cities.length;i++)

{

System.***out***.printf("%-10d %-13s %n",i,cities[i]);

}

System.***out***.printf("City Count: %d",cityCount);

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

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

}

**else** **if**(choice==9)

{

System.***out***.println("Exiting Program. Thank you");

}

**else**

{

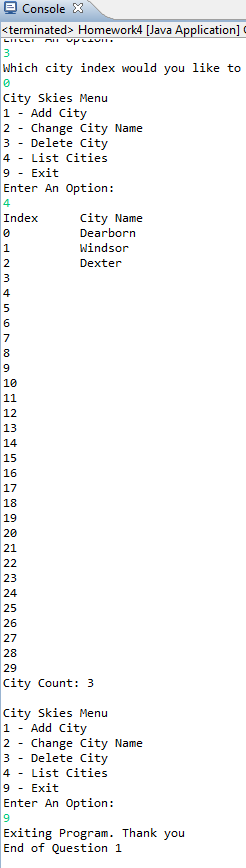
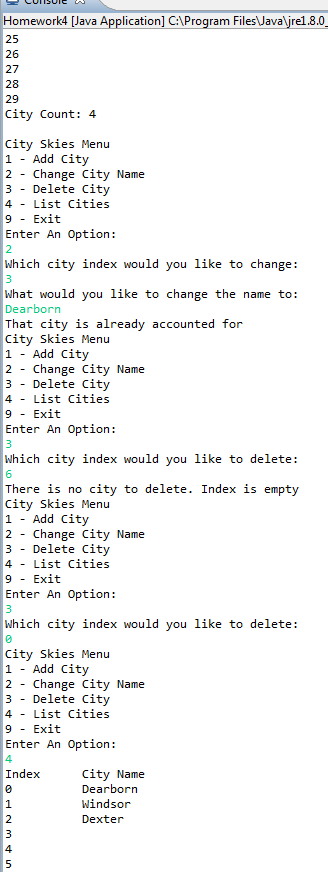
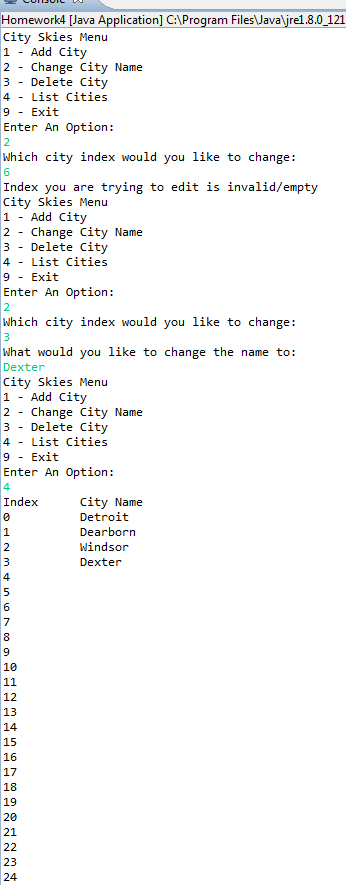
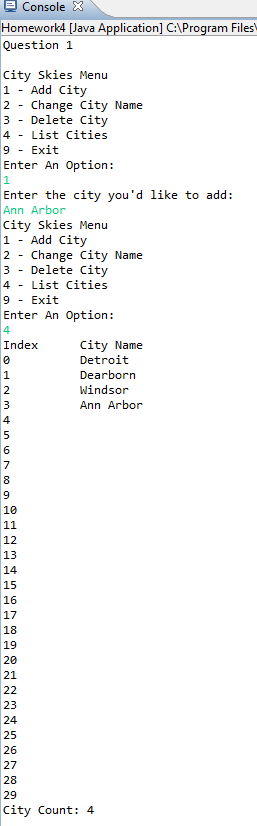
System.***out***.println("Invalid Option. Please Try Again");

}

}

System.***out***.println("End of Question 1");

*[your program output here]\*\**

**

**2) [10 points]** You've been hired by *Al’s Algorithms* to write a Java console application that performs a selection sort on an integer array list. Create the array list and add twenty random integers to it. Generate the random integers between 100 and 200. Print the array list, sort it using a selection sort, and print the array list again.

*[your program code here]\**

**package** homework\_4;

**import** java.util.Scanner;

**public** **class** Homework4 {

**static** Scanner *keyboard*=**new** Scanner(System.***in***);

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

System.***out***.println("Start of Question 2");

**int** temp; //variable used for temporary storing variable to be switched

**int**[] sortArray=**new** **int**[20];

**for**(**int** i=0;i<sortArray.length;i++) //populate the array with random variables

{

sortArray[i]=100+(**int**)(Math.*random*()\*((200-100)+1)); //this function was found on stackoverflow.com

//http://stackoverflow.com/questions/363681/generating-random-integers-in-a-specific-range

}

System.***out***.println("Unsorted Array");

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

**for**(**int** i=0;i<sortArray.length;i++) //print unsorted array

{

System.***out***.println(sortArray[i]);

}

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

**for**(**int** i=0;i<sortArray.length;i++) //sort the array

{

**for**(**int** j=0;j<sortArray.length;j++)

{

**if**(sortArray[j]<sortArray[i])

{

temp=sortArray[i];

sortArray[i]=sortArray[j];

sortArray[j]=temp;

}

}

}

System.***out***.println("Sorted Array. Decending Order");

**for**(**int** i=0;i<sortArray.length;i++)

{

System.***out***.println(sortArray[i]);

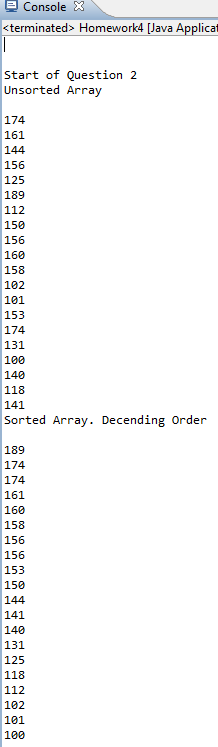
}

*keyboard*.close();

}

}

*[your program output here]\*\**

**

\* **Copying-and-pasting application code to a Word document**

1) From the program editor window, press **CTRL-A** and press **CTRL-C**.

2) From within the Word document, press **CTRL-V**.

\*\* **Copying-and-pasting application output to a Word document**

1) From the Eclipse main screen, maximize the Console window.

2) From the Console window, press **ALT-PrintScreen**.

3) From within the Word document, press **CTRL-V**.