King Saud University
College of Computer & Information Science
CSC111 – Tutorial13
Arrays – II –
All Sections

Objectives:

- To know how to define and create an array.
- To know how to access array elements.
- To know how to iterate over arrays using loops
- To know how to manipulate arrays
- To know how to add elements to array and delete them
- To know how to search arrays
- To know how to send arrays to methods

(***Exercise with * can be left to the student as self review questions***)

Exercise 1

1. Write a method **add** that receives an array of integers **arr**, the number of the elements in the array **arr** and an integer **n**. It then adds the integer **n** to the array **arr** if the number of elements in the array is less than its size. Method **add** uses another method **find** that checks if the integer **n** is in the array or not. Method **add** returns **false** if **n** can not be added or is already in **arr**.

2. Write a method **flipCoin** that receives an array of boolean **flips** and the number of coin flips so far. The method randomly flips a coin by calling method **nextBoolean** of class **java.util.Random** and stores the new flip in array **flips** if array is not full.

3. Write a method **deleteTweet** that receives your **tweets**, their number and a tweet that you would like to remove. The method then searches for the tweet and delete it from your twitter history. If tweet was not found, an error message is reported.

4. Write a method **findMove** that receives the history of moves made by a robot, the number of moves so far and a move. A move consists of two parts **dx** and **dy** which represent the amount of units traveled on x and y axis. The history is stored in two arrays one for each axis. The method looks up the move and returns its index in the two arrays otherwise it returns -1.

Solution

1)

```
public int find(int[] arr, int num, int n){
	for (int i = 0; i < num; i++) {
		if (arr[i] == n) {
			return i;
		}
	}
	return -1;
}

public boolean add (int[] arr, int num, int n) {
	if (num < arr.length) {
		if (find(arr, num, n) != -1){
			arr[num] = n;
			num++;
			return true;
		}
		else
			System.out.println("ERROR: ELEMENT ALREADY"
						+ " ADDED.");
	} else
		System.out.println("ERROR: ARRAY IS FULL");
	return false;
}
```

2)

```
public void flipCoin (boolean[] flips, int num) {
	if (num < flips.length) {
		java.util.Random r = new java.util.Random();
		boolean newFlip = r.nextBoolean();
		flips[num] = newFlip;
		num++;
	} else
		System.out.println("ERROR: CAN NOT FLIP COIN");
}
```

3)

```
    public void deleteTweet(String[] tweets, int numOfTweets,
                            String tweet) {
          boolean found = false;
          for (int i = 0; i < numOfTweets && !found; i++) {
                if (tweets[i].equalsIgnoreCase(tweet)) {
                      tweets[i] = tweets[numOfTweets];
                      found = true;
                }
          }
          if (!found)
                System.out.println("ERROR: TWEET IS "
                                  + "ALREADY DELETED");
    }
```

4)

```
    public int findMove(double[] xMoves, double[] yMoves,
                double dx, double dy, int numMoves) {
          for (int i = 0; i < numMoves; i++)
                if ((xMoves[i] == dx) && (yMoves[i] == dy))
                      return i;
          return -1;
    }
```

Done…