

Seminar 4 (Arrays, Methods) Exercises

1. The day of a week is declared in an array as follows in a program:

```
String[] days = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday",  
                "Saturday", "Sunday"};
```

- a. Write a program that has 1 integer parameter value and prints the word equivalent of the day. Assume the number will be between 1-7, 1 representing Monday. You should not use any if...else statements.
- b. Print a table of the days of the week as follows:
 1 Monday
 2 Tuesday
 ...
 7 Sunday

```
public class Q1  
{  
    public static void main(String[] args){  
        String[] days = {"Monday", "Tuesday", "Wednesday", "Thursday",  
"Friday", "Saturday", "Sunday"};  
        //part a  
        int n = Integer.parseInt(args[0]);  
        System.out.println( days[n] );  
  
        //part b  
        for ( int d=0; d<7; d++)  
        {  
            System.out.println( (d+1) + " " days[d]);  
        }  
    }  
}
```

2. Declare an integer array of size 5.

Write separate **loop** structures for each of the following:

- a. Read (using Scanner class) in 5 values into each element of the array.

Example of input is as follows:

Enter value 1: 99

Enter value 2: 88

...

- b. Increment all the elements of the array by 1.
- c. Display the smallest number.
- d. Display the sum of all the numbers in the array.
- e. Search for a number in the array. If found, display a message that the number is found.
- f. Declare another integer array y of the same length as x. Copy the values of each element in x to y. Display the elements in y to confirm it has been copied correctly.

```

import System.util.Scanner;
public class Q2
{
    public static void main(String[] args){
        int[ ] x = new int[5];

        //a. Read in
        Scanner console = new Scanner(System.in);
        for (int n=0; n<x.length; n++)
        {
            System.out.print("Enter value " + (n+1) + ": ");
            x[n] = console.nextInt();
        }

        //b. increment all elements by one
        for ( int i=0; i< x.length; i++)
            x[i]++;

        //c. Display the smallest number
        int smallest = x[0]; //assume first element smallest
        for ( int i=1; i<x.length; i++)
            if ( x[i] < smallest)
                smallest = x[i];
        System.out.println("The smallest number is " + smallest);

        //d. Display the sum
        int sum=0;
        //set up a loop to add the sum
        for ( int i=0; i< x.length; i++)
            sum += x[i];
        System.out.println("Sum = " + sum);

        //e. Search the array for a number
        int searchFor=4; //assume looking for 4
        boolean found = false;
        for ( int i=0; i<x.length; i++)
            if ( x[i] == searchFor ){
                found = true;
                break;
            }
        if ( found )
            System.out.println("Found!");
        else
            System.out.println("Not found!");

        //f. Copy to another array
        int[] y = new int[x.length];
        //set up a loop and copy from x[i] to y[i];
        // something like this: y[i]=x[i];
        for ( int i=0; i<x.length; i++)
            y[i]=x[i];
    }
}

```

3. A test consists of 10 MCQ questions. Each question has 4 choices: a, b, c, d. The answers to each question is stored in an array as follows:
- ```
char[] mcq = { ' ', 'a', 'b', 'b', 'a', 'd', 'c', 'b', 'a', 'b', 'c' };
```
- That means the answer to question 1 is a, question 2 is b, etc.

Write a program that uses the above array declaration and prompts for the answer to the 10 MCQ questions. The results are displayed only after answers to all the questions are completed. Details of the answers as well as a count of the correct answers are then displayed. An input sample is as follows:

Q1: **a**  
Q2: **b**  
Q3: **c**  
...  
Q10: **d**

Output:

Q1: a correct  
Q2: b correct  
Q3 c incorrect, answer is b  
...  
Q10: d incorrect, answer is c  
Total 7 out of 10 correct.

```
import java.util.Scanner;

public class Q3
{
 public static void main(String[] args)
 {
 char[] mcq={' ', 'a', 'b', 'b', 'c', 'd', 'c', 'b', 'a', 'b', 'c'};
 char[] ans = new char[11];
 Scanner console = new Scanner(System.in);
 for(int n=1; n<=10; n++)
 {
 System.out.print("Q + n + ": ");
 ans[n] = console.next().charAt(0);
 }
 int correct = 0;
 for (int n=1; n<=10; n++)
 {
 if (ans[n]==mcq[n])
 {
 System.out.println(ans[n] + " correct");
 correct++;
 }
 else
 System.out.println(ans[n]+" incorrect, answer is "
+mcq[n]);
 }
 System.out.println("Total " + correct + " out of 10 correct");
 }
}
```

4. Trace the output of the following:

```
int[] x = {1, 2, 3, 4, 5};
```

```
for (int i=0; i<x.length-1; i++){
 x[i+1]=x[i+1] + x[i];
}
for (int i=0; i<x.length; i++){
 System.out.print(x[i] + " ");
}
```

|             |
|-------------|
| 1 3 6 10 15 |
|-------------|

5. a. Write a static method max that has 2 double parameters. The method returns the larger of the 2 values. Test the method.
- b.. Write a static method max that has 3 double parameters. The method returns the largest of the 3 values. Make use of the max method in part a. Test the method.

```
public class Q5
{
 public static void main(String[] args)
 {
 System.out.println(max(1,2));
 System.out.println(max(1,2,3));
 }

 //part a
 public static double max(double n1, double n2)
 {
 if (n1 > n2)
 return n1;
 return n2;
 }

 //part b
 public static double max(double n1, double n2, double n3)
 {
 return max(max(n1, n2), n3);
 }
}
```

6. Write a static method `getGrade` that has 1 double parameter representing a mark of an assessment. The method returns a String as follows:

Distinction – 75 and above

Credit – 50 to less than 75

Fail – less than 50

Write a main method to test the `getGrade` method.

```
public class Q6
{
 public static void main(String[] args)
 {
 System.out.println(getGrade(75));
 }

 public static String getGrade(double mark)
 {
 if (mark >= 75)
 return "Distinction";
 else if (marks >=50)
 return "Credit";
 else
 return "Fail";
 }
}
```

- 7 a. Write a static method reverse that has a String parameter. The method returns the reverse of the string. E.g. if "abcde" is passed to the method, it returns "edcba". Test the method.
- b. Write a static method isPalindrome that has a String parameter. The method returns true if the string is a palindrome. A palindrome is a string that reads the same in reverse. For example, "anna" is a palindrome. Make use of the reverse method in part a.
- c.. Write statements to test the isPalindrome method.

```
public class Q7
{
 public static void main(String[] args)
 {
 //part c
 String s = "hannah";
 if (isPalindrome(s))
 System.out.println(s + " is a Palindrome");
 else
 System.out.println(s + " is not a Palindrome");
 }

 //part a
 public static String reverse(String s)
 {
 String temp="";
 for (int n=s.length()-1; n>=0; n--)
 temp += s.charAt(n);
 return temp;
 }

 //part b
 public static boolean isPalindrome(String s)
 {
 if (reverse(s).equals(s))
 return true;
 return false;

 //or use this:
 return reverse(s).equals(s);
 }
}
```



8. Reorganize each part of Q2 using method calls.

- a. Use a method `populateArray` that has the array as parameter. It populates the array as described in Q5a. No return values.
- b. Use a method `incrementArray` that has 2 parameters – the array and a number that indicates the value to increment for each element of the array. No return values.
- c. Use a method `computeTotal` that has 1 parameter – the array. The method sums all the elements of the array and returns this value.
- d. Use a method `searchArray` that has 2 parameters – the array and a number to search. The method returns true if the number can be found in the array and false otherwise.
- e. Use a method `findSmallest` that has 1 parameter – the array. It returns the smallest number in the array.
- f. Use a method `copyArray` that has 2 parameters – 2 arrays. It copies the values in the first array to the second. No return value.

```
import System.util.Scanner;
public class Q8
{
 public static void main(String[] args){
 int[] x = new int[5];

 //a. Read in
 populateArray(x);

 //b. increment all elements by one
 incrementArray(x, 1);

 //c. Display the sum
 System.out.println("Sum = " + computeTotal(x));

 //d. Search the array for a number
 if (searchArray(x, 5))
 System.out.println("Found!");
 else
 System.out.println("Not found!");

 //e. Display the smallest number
 int smallest = findSmallest(x);
 System.out.println("The smallest number is " + smallest);

 //f. Copy to another array
 int[] y = new int[x.length];
 copyArray(x, y);
 }

 public static void populateArray(int[] x)
 {
 Scanner console = new Scanner(System.in);
 for (int n=0; n<x.length; n++)
 {
 System.out.print("Enter value " + (n+1) + ": ");
 x[n] = console.nextInt();
 }
 }
}
```

```

public static void incrementArray(int[] x, int n)
{
 for (int i=0; i< x.length; i++)
 x[i]+=n;
}

public static int computeTotal(int[] x)
{
 int sum=0;
 //set up a loop to add the sum
 for (int i=0; i< x.length; i++)
 sum += x[i];
 return sum;
}

public static boolean searchArray(int[] x, int n)
{
 for (int i=0; i<x.length; i++)
 if (x[i] == n){
 return true;
 }
 return false;
}

public static int findSmallest(int[] x)
{
 int smallest = x[0]; //assume first element smallest
 for (int i=1; i<x.length; i++)
 if (x[i] < smallest)
 smallest = x[i];
 return smallest;
}

public static void copyArray(int[] x, int[] y)
{
 //set up a loop and copy from x[i] to y[i];
 // something like this: y[i]=x[i];
 for (int i=0; i<x.length; i++)
 y[i]=x[i];
}
}

```

9. Trace the output of the following program:

```
public static void main(String[] args)
{
 int[] x = {1, 2, 3, 4, 5};
 int[] y = new int[5];

 for (int i=0; i<x.length; i++){
 y[i] = methodA(x[i]);
 }

 methodB(y);
 for (int i=0; i<x.length; i++){
 System.out.print(y[i] + " ");
 }
}

public static int methodA(int x){
 return x * 2;
}

public static void methodB(int[] x){
 for (int i=0; i<x.length; i++)
 x[i] += 1;
}
```

Output:

3 5 7 9 11

10. A check digit is usually appended to a code number in order to detect errors arising when the number is transcribed manually.

The check digit of the NRIC No. is the official reference which is determined as follows:

For example : NRIC No.(with official reference) = S 7928964 G

Step 1:

Multiply each digit by the following weights.

|            |    |    |    |    |    |    |   |
|------------|----|----|----|----|----|----|---|
| NRIC No. : | 7  | 9  | 2  | 8  | 9  | 6  | 4 |
| Weights :  | 2  | 7  | 6  | 5  | 4  | 3  | 2 |
| Products:  | 14 | 63 | 12 | 40 | 36 | 18 | 8 |

Step 2:

Sum the products of each digit x weight.

Sum :  $14 + 63 + 12 + 40 + 36 + 18 + 8 = 191$

Step 3:

Find the remainder when the sum is divided by 11.

Sum/11: 17 remainder 4

Step 4:

Take 11 – remainder to get the check digit.

Check digit :  $11 - 4 = 7$

Step 5:

Look up the following table to get the official reference.

Official Reference : G

|             |   |   |   |   |   |   |   |   |   |    |    |
|-------------|---|---|---|---|---|---|---|---|---|----|----|
| Conversion: | A | B | C | D | E | F | G | H | I | Z  | J  |
| Table       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |

Write a method `getReference` that has a `String` parameter representing the NRIC( E.g. "1234567"). The method returns the official reference as a `char`.

Test the method.

```

public class Q10
{
 public static void main(String[] args)
 {
 System.out.println(getReference("1234567"));
 }

 public static String getReference(String nric)
 {
 int[] weight = { 2,7,6,5,4,3,2};
 char[] ref = {' ', 'A','B','C','D','E','F','G','H','I','Z','J'};

 //Step 1 & 2: multiply and sum each digit by weight

 int sum=0;
 for(int i=0; i<nric.length(); i++)
 sum += weight[i] * Integer.parseInt(""+nric.charAt(i));

 // ""+nric.charAt(i) is to make it a String so that we can use
 parse it

 //Step 3 & 4: Find remainder and get check digit
 int rem = sum % 11;
 //Step 5: Look up table for reference
 return ref[rem];
 }
}

```