King Saud University
College of Computer & Information Science
CSC111 – Lab11
Array of objects
All Sections

Instructions

Web-CAT submission URL:
http://10.131.240.28:8080/Web-CAT/WebObjects/Web-CAT.woa/wa/assignments/eclipse

Objectives:

- To know how to define and create an array.
- To know how to pass array to method and return array from method.
- To know how to create object with arrays as attributes
- To know how to add elements to arrays
- To know how to search arrays
- To know how to find max element in an array

Lab Exercise 1 (Lab Homework) – Expected Time: 2 hours

Part 1 (read/write array values)

Write a program **CourseManager1** that reads and prints scores of students in a course. The scores are double numbers between 0 and 100.

- Your program should start by reading the number of the students taking the course.
- Then it reads and stores the scores in an array. If a score is invalid then your program should store 0.
- After that it prints the scores.

Sample Run

```
Enter number of students: 4 ↵
Please enter students' scores: 100 -30 75 90 ↵
The score -30.0 you entered is wrong. Program will store score 0.
The scores are: 100.0 0.0 75.0 90.0
```

Complete following pseudo code

```
import java.util.Scanner;
public class CourseManager1 {
	public static void main(String[] args)
	{
		Scanner input = new Scanner(System.in);
		System.out.print("Enter number of students: ");
		// read the number of students ( array size)

		while (/* array size less than 1*/numOfStudents < 1){
			System.out.print("Number of students is invalid. Enter
number of students: ");
			// read array size  again

		}
		// define and declare the array ( double)

		System.out.print("Please enter students' scores: ");
		//read the array score
		for (int i = 0; i < scores.length; i++){

			//read the array score
			if (score >= 0 && score <= 100){
				scores[i] = score;
			}
			else {
```

```
                        System.out.println("The score " + score + " you
entered is wrong. Program will store score 0.");
                    }
            }
            System.out.print("The scores are: ");


            // write a code to output the contents of the array (print the
array score)
            System.out.println();
      }// end of main method

}// end of CourseManager1 class
```

At this point, submit your program to WebCAT.

Part 2 (copy array, pass array as parameter, return array as result)

Modify the previous program such that after reading the scores, your program computes the letter grades and store them in an array of type `char`.

- Write a static method `scoreToGrade` that
  - takes the **scores** array as parameter,
  - creates the **grades** array,
  - fill **grades** array up with letter grades (A: 90-100, B: 80-89, C: 70-79, D: 60-69, F: 0-59) then return it.

  Use the rules of KSU to convert a score into a letter grade.
- Prints each score along with the letter grade using format *score/letter_grade*.

Name your new program `CourseManager2`.

Sample Run

```
Enter number of students: 5 ↵
Please enter students' scores: 100 40 79 89 90 ↵
The scores/grades are: 100.0/A 40.0/F 79.0/C 89.0/B 90.0/A
```

Complete following pseudo code

```
import java.util.Scanner;
public class CourseManager2 {
```

```
	public static void main(String[] args) {
		Scanner input = new Scanner(System.in);
		System.out.print("Enter number of students: ");
		// read the number of students ( array size)
		while (/* array size less than 1*/){
			System.out.print("Number of students is invalid. Enter
number of students: ");
			// read array size  again

		}
// define and declare the  scores array ( double)

		System.out.print("Please enter students' scores: ");

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

			//read score and store it in the array

		}
		// define and declare the  grades array ( char)
		char[] grades = scoreToGrade(scores);
		System.out.print("The scores/grades are: ");
		for (int i = 0; i < scores.length; i++){
			System.out.print(scores[i] + "/" + grades[i] + " ");
			// prints each score along with the letter grade using
format scores[i] + "/" + grades[i] + " "

		}
		System.out.println();
	}

	// Precondition: all scores in the array are between 0 and 100
	// create scoreToGrade method
	public static char[] scoreToGrade(double[] scores){
		char[] grades = new char[scores.length];
		for (int i = 0; i < scores.length; i++){
			if (scores[i] >= 90)

			//if score >=90 store A in grade

			//if score > 80 store B in grade

			//if score > 70 store C in grade
			else if (scores[i] >= 60)
				grades[i] = 'D';
			//if score > 60 store D in grade
			else
				// store F
				grades[i] = 'F';
		}

		return grades;
	}

}
```

At this point, submit your program to WebCAT.

Part 3

Since we are doing object oriented programming, a better design is to declare and use arrays as attributes of the class **CourseManager**. This means that we will avoid passing/return arrays to/from methods.

Rewrite previous program using object oriented programming methodology as following:

- Define the two arrays `scores` and `grades` as attributes of the class `CourseManager3`.
- Change the method `scoreToGrade` such that:
  - It becomes an instance method.
  - It does not receive or return anything.
- Add a methods `readScores` to create the `scores` array and read its values.
- Add a methods `printGrades` to print the `scores` and `grades` arrays as done in previous program.

Create a program `TestCourseManager3` that does exactly the same as previous program but using an instance of class `CourseManager3`.

Sample Run

```
Enter number of students: 5 ↵
Please enter students' scores: 100 40 79 89 90 ↵
The scores/grades are: 100.0/A 40.0/F 79.0/C 89.0/B 90.0/A
```

Complete following pseudo code

```
import java.util.Scanner;

class CourseManager3 {
	//declare an array scores ( double)
	//declare an array score (char)
```

```
	//create a method readScores()
	public void readScores()
	 {
		Scanner input = new Scanner(System.in);
		System.out.print("Enter number of students: ");
		int numOfStudents = input.nextInt();
		while (/* array size less than 1*/) {
			System.out.print("Number of students is invalid. Enter
number of students: ");
			//READ numOfStudents  AGAIN
		  }

		scores = new double[numOfStudents];
		System.out.print("Please enter students' scores: ");
		for (int i = 0; i < scores.length; i++)
		 {
			double score = input.nextDouble();

		// print a message ""The score " + score + you entered is wrong.
Program will store score 0." if you entered a wrong score.

		}
	   }

	// Precondition: all scores in the array are between 0 and 100
	public void scoreToGrade() {
		//create a method scoreToGrade with return type void

		// define array grades with type char
		for (int i = 0; i < scores.length; i++) {

			//if score >=90 store A in grade
			else if (scores[i] >= 80)
				grades[i] = 'B';
			//if score > 80 store B in grade

			//if score > 70 store C in grade

			//if score > 60 store D in grade
			else
				grades[i] = 'F';
			// store F
		}
	}

	public void printGrades(){
		//create a method printGrades with return type void

		}
		System.out.println();
	}
}
 //create a class TestCourseManager3

public class TestCourseManager3 {
```

```
	public static void main(String[] args) {
		CourseManager3 cm = new CourseManager3();
		//create an object cm of Class CourseManager3 and call
readScores, scoreToGrade and printGrades


	}}
```

At this point, submit your program to WebCAT.

Part 4

Modify previous program by adding two methods to class `CourseManager4` that will compute the average score of the course. Methods are:

- `sumScores` which computes and returns the sum of scores in `scores` array. This method is an *internal helper method* and it should be `private`. It will be used by the next method `average`.
- `average` which computes and returns the average of scores in `scores` array using `sumScores` as an *internal helper method*.

Now write the main program to do the same as previous one in addition to printing the scores average. Name your program `TestCourseManager4`.

Sample Run

```
Enter number of students: 5 ↵
Please enter students' scores: 100 40 79 89 90 ↵
The scores/grades are: 100.0/A 40.0/F 79.0/C 89.0/B 90.0/A
Average = 79.6
```

Complete following pseudo code

```
import java.util.Scanner;

class CourseManager4 {
	//declare an array scores ( double)
	//declare an array score (char)
```

```
	public void readScores()
	 {
		Scanner input = new Scanner(System.in);
		System.out.print("Enter number of students: ");
		int numOfStudents = input.nextInt();
		while ((/* array size less than 1*/))
		 {
			System.out.print("Number of students is invalid. Enter
number of students: ");
			numOfStudents = input.nextInt();
		}
		scores = new double[numOfStudents];
		System.out.print("Please enter students' scores: ");
		for (int i = 0; i < scores.length; i++) {
				// print a message ""The score " + score + you
//entered is wrong. Program will store score 0." if you entered a wrong score.
		}
	}

	// Precondition: scores is not null and all scores in the array are
between 0 and 100
	public void scoreToGrade() {
		grades = new char[scores.length];
		for (int i = 0; i < scores.length; i++) {
			//if score >=90 store A in grade
			else if (scores[i] >= 80)
				grades[i] = 'B';
			//if score > 80 store B in grade

			//if score > 70 store C in grade

			//if score > 60 store D in grade
			else
				grades[i] = 'F';
			// store F
		}
	}

	// Precondition: scores and grades are not null
//CREATE A printGrades method.
	public void printGrades()
	{
	}

//create a method sum() with return type double which should return sum of all
the elements of an array scores[]
	private double sum()
	{

	}

	// Precondition: scores is not null
	//create a method average which has return type double ( formula sum() /
scores.length)
	public double average(){
```

```
	}
}

public class TestCourseManager4 {
	public static void main(String[] args) {
		CourseManager4 cm = new CourseManager4();
		//create an object cm of Class CourseManager4 and call
readScores, scoreToGrade, printGrades and average

	}

}
```

At this point, submit your program to WebCAT.

Small exercise: change method `average` to `averageScore` and create a new method `averageGrade` that returns the average letter grade based on the average score.

Part 6

Common mistakes. If you have noticed, many methods in the class `CourseManager` has a comment `//precondition`. Why? Why `sumScores` does not have such comment? (Hint: think what will happen if the user of class `CourseManager` calls `printGrades` before `readScores`).

Lab Exercise 2 – Expected Time: 01:50 hours

In this exercise, we will make major changes to previous program to make it an interactive course manager.

Part 1 (add elements to array) – Expected Time: 50 min

Write a class **`CourseManager5`** that stores a list of students' objects in a given class. Each student has an id, name, score. The class allows the user to add a student, and display students' data. Here is the UML diagram:

TestCourseManager5
+ main()

Student
-id: int -name: String -score: double
Student(id:int, name:String, score, double) +getId():int +getName():String +getScore():double

CourseManager5
-students[]: Student -nStudents: int +MAX_SIZE: int
CourseManager5() +getNStudents():int +addStudent(newStudent: Student): void +dispalyStudent(i: int): void

As shown in the UML diagram, write the class **`Student`** that has the attributes: ids, names, scores, and a constructer to initialize the attributes. Then write the class **`CourseManager5`** that has an array of student objects. The attribute `nStudents` represents the current number of students in the list. *The maximum number of students in the class is 100*.

The methods are:

- `CourseManager5`: a constructor that initializes the attributes and creates an array of students of size 100.
- `getNStudents` : returns the current number of students.
- `addStudent`: adds the student with the given object to the list. If course is full, it prints the error message: “ERROR: COURSE IS FULL” .
- `displayStudent`: displays all data of the student at index `i`.

Write a main class called **`TestCourseManager5`** with a `main` method that will do the following:

. It creates a `CourseManager5` object.

. Then, it adds 3 students by reading their IDs, names, and scores from the user.

. Then, it displays all students in class.

Sample run

```
Please enter the ID, name, and score of student 0:
434000000 ↵
Ahmed ↵
95 ↵
Please enter the ID, name, and score of student 1:
433001234 ↵
Ali ↵
85 ↵
Please enter the ID, name, and score of student 2:
434005421 ↵
Fahad ↵
76 ↵
Students are:
434000000, Ahmed, 95.0
433001234, Ali, 85.0
434005421, Fahad, 76.0
```

Complete following pseudo code

```
class CourseManager5
     /* Declare the class data members as shown in the UML*/

    CourseManager5() {
           /* Write the constructor that initializes the
           attributes and creates the array of students of size
           100 and nStudents should be initilized to 0. here */
    }

    public /* method modifier*/ addStudent(/* student object */)
{

            /* 1- check if nStudents is less than the maximum
           size to add a new student else print the message:
                       System.out.println("ERROR: COURSE IS
FULL");*/

                       /* 2- add the new student to the list
           Increment the current number of students*/
    }

    public /* method modifier */ displayStudent(/* parameters
list */) {
                      /* print the id, name , and scores of the
index i passed to the method */
    }

    public /* method modifier */ getNStudents() {
                  /* retun nStudents */
     }
}


import java.util.Scanner;
public class TestCourseManager5 {
    public static void main(String[] args) {
     Scanner kb = new Scanner(System.in);

             /* creat a CourseManager5 object named c1*/

             /* use for loop to do the following 3 times:

             1- ask the user to enter student information ID,
     name, and score */
```

```
            //System.out.println("Please enter the ID, name,
      and score of a student: ");

            /* 2- use the scanner to get the id, name , and
      score */

              /* 3- create a student with the new information"
      */
           /* 4- add student using the method "addStudent" */


            /* display all students in class. */
      }
}
```

Part 2 (find elements in an array) – Expected Time: 30 min

Modify previous program by adding a method to find a student by name. Name your new class **`CourseManager6`**. Modify `addStudent` such that it uses the `findStudentByName` method to make sure the student is not added twice to class. Here is the UML diagram:

TestCourseManager6
+ main()

CourseManager6
-students[]: Student -nStudents: int +MAX_SIZE: int
CourseManager6() +getNStudents():int +addStudent(newStudent: Student): void +findStudentByName(name: String) : int +dispalyStudent(i: int): void

As shown in the UML diagram, the new and modified methods are:

- `addStudent`: adds the student with the given students' object to the list. If course is full, it prints the error message: "ERROR: COURSE IS FULL". If student is already added it prints the error message: "ERROR: STUDENT ALREADY ADDED".

- `findStudentByName`: returns the index of the student whose name is `name`. If it is not found, -1 is returned.

Write a main class called **`TestCourseManager6`** with a `main` method that will do the following:

. It creates a `CourseManager6` object.

. Then, it adds a student by reading its ID, name, and score from the user.

. Then, it tries to add the same student again and prints a failure message.

. Then, it displays the students.

Sample run

```
Please enter the ID, name, and score of a student:
434001234 ↵
Ahmed ↵
65 ↵
Please enter the ID, name, and score of a student:
434001234 ↵
Ahmed ↵
65 ↵
ERROR: STUDENT ALRAEDY THERE
Students are:
434001234, Ahmed, 65.0
```

Complete following pseudo code

```
class CourseManager6 {
/* Declare the class data members as shown in the UML*/

    public CourseManager6() {
            /* Write the constructor that initializes the
           attributes and creates the array of students of size
           100. here */
    }

    public /* method type */ addStudent(/* students' object */) {
            /* 1- check if nStudents is less than the maximum
           size to add a new student else print the message:
                System.out.println("ERROR: COURSE IS FULL");*/

             /* 2- check if the student is not already in the
           list by using the methos findStudentName
                 if the student is not in the list, add the new
```

```
student.
                if the student is already in the list print
                 System.out.println("ERROR: STUDENT ALRAEDY
THERE");
                */
    }

    public /* method type */ findStudentName(/* parameters list
*/) {

                /* 1- use for loop to check the student list */
             /* 2- check if the name in the array "students[]" is
            equal to the name passed to the method
                  if you find the name return the index number
                  otherwise return -1 */
    }

    public /* method type */ displayStudent(/* parameters list
*/) {

                        /* print the id, name , and scores of the
index i passed to the method */
    }

    public /* method type */ getNStudents(/* parameters list */)
{

                        /* retun nStudents */
    }
}


import java.util.Scanner;
public class TestCourseManager6 {
    public static void main(String[] args) {
      Scanner kb = new Scanner(System.in);

            /* creat CourseManager6 object named c1 */

             /* ask the user to enter student information ID,
    name, and score */
          //System.out.println("Please enter the ID, name, and
    score of a student: ");

            /* use the scanner to get the id, name , and score */

            /* create a student with the new information. */

            /* add student using the method "addStudent" */

            /* ask the user to enter another student information
```

```
ID, name, and score */
        //System.out.println("Please enter the ID, name, and
score of a student: ");
      /* - create a students' object with the new
information" */

         /* add the student using the method "addStudent" */

        //System.out.println("Students are: ");

        /* print the student list using for loop and the
methode displayStudents */

  }
}
```

Part 3 (find max element in an array) – Expected Time: 30 min

Modify previous program by adding two methods to find the student with maximum score and compute the average score. Name your new class **`CourseManager7`** and add methods `findMaxScoreIndex` and `findAverageScore` to the class. Here is the UML diagram:

TestCourseManager7
+ main()

CourseManager7
-students[]: Student -nStudents: int +MAX_SIZE: int

```
CourseManager7()
+getNStudents():int
+addStudent(newStudent: Student): void
+findStudentByName(name: String) : int
+dispalyStudent(i: int): void
+findMaxScoreIndex() : int
+findAverageScore(): double
```

As shown in the UML diagram, the new and modified methods are:

- `findMaxScoreIndex`: returns the index of a student whose score is the highest in the class.
- `findAverageScore`: returns the average score of the class.

Write a main class called **`TestCourseManager7`** with a `main` method that will do the following:

. It creates a `CourseManager7` object.

. Then, it adds 3 students by reading their IDs, names, and scores from the user.

. Then, it displays the average class scores.

. Then, it displays the student with the maximum score.

Sample run

```
Please enter the ID, name, and score of student 0:
433000111 ↵
Mohammad  ↵
60.0      ↵
Please enter the ID, name, and score of student 1:
433000222 ↵
Ahmad     ↵
100.0          ↵
Please enter the ID, name, and score of student 2:
433000333 ↵
Khalid    ↵
50.0      ↵
The class average = 70.0
The student with the highest score:
```

433000222, Ahmad, 100.0

Complete following pseudo code

```
class CourseManager7 {
        /* Declare the class data members as shown in the UML or
    copy from CourseManager6 and paste here*/

    public CourseManager7() {
      /* copy from CourseManager6 and paste here* */
         }

     public /* method type */ addStudent(/* students' object */)
{

              /* copy from CourseManager6 and paste here*   */
    }


    public /* method type */ findStudentName(/* parameters list
*/) {

             /* copy from CourseManager6 and paste here */
    }

    public /* method type */ findAverageScore(/* parameters list
*/) {
            /* 1- check if nStudents is greater than zero which
    means that the list is not empty
             if the list is empty return 0 */

            /* 2- use for loop to calculate the sum of all scores
    */

            /* 3- return  the average ==> sum/nStudents */
     }

    public /* method type */ findMaxScoreIndex(/* parameters list
*/) {

            /* create integer max = 0;

            /* 2- check if nStudents is greater than zero which
    means that the list is not empty

             if nStudents is equla to or less than 0  make max =
    -1 */

            /* 3- use for loop to compare the score of every
    student to the max score
```

```
                3.1- the max score should be initialized to be
    equal to the first element
     max = 0;

              the comparison should be like this if (score
    [i]>score[max]) => max = i; */

            /* 4- return max */
}

    public /* method type */ displayStudent(/* parameters list
*/) {

            /* print the id, name , and scores of the index i
passed to the method */
    }

    public /* method type */ getNStudents(/* parameters list */)
{

                /* return nStudents */
    }
}

import java.util.Scanner;
public class TestCourseManager7 {
    public static void main(String[] args) {
      Scanner kb = new Scanner(System.in);

        /* creat CourseManager7 object named c1 */

         /* use for loop to do the following 4 times:

         /* 1- ask the user to enter student information ID, name,
and score */
         //System.out.println("Please enter the ID, name, and
score of a student: ");

         /* 2- use the scanner to get the id, name , and score */

    /* 3- create a student with the new information. */

         /* 4- add student using the method "addStudent" */

         /* after the loop print the class average score */
         /* print the student information whos has the max score
using the method displayStudent */
       }}
```

Part 4 (Delete an element from an array)

Modify previous program by adding two methods to remove an elemet from an array. Name your new class **`CourseManager8`** and add methods `removeStudent` and `removeAndShiftStudents` to the class. Here is the UML diagram:

TestCourseManager8
+ main()

CourseManager8
-students[]: Student -nStudents: int +MAX_SIZE: int
CourseManager7() +getNStudents():int +addStudent(newStudent: Student): void +findStudentByName(name: String) : int +dispalyStudent(i: int): void +findMaxScoreIndex() : int +findAverageScore(): double +removeStudent(index: int) +removeAndShiftStudents(index: int)

As shown in the UML diagram, the new and modified methods are:

- `removeStudent`: removes an Student with the given index and setting its value to NULL.
- `removeAndShiftStudents`: you shift elements with index greater than i left by one element. For example, if you want to **remove** element 3, you copy element 4 to element 3, element 5 to element 4 etc.

Write a main class called **`TestCourseManager8`** with a `main` method that will do the following:

. It creates a `CourseManager8` object.

. Then, it adds 3 students by reading their IDs, names, and scores from the user.
. Then, it removes an element at `index` 1 using `removeStudent` method.
. Then, it displays the students make sure to handle to the `null` object before you print it.
. Rerun the program again and instead of calling `removeStudent` method call `removeAndShiftStudents` to remove an element at index 0.
. Then, print the arrays' content to see the change.

Sample run using `removeStudent`

```
Please enter the ID, name, and score of student 0:
433000111 ↵
Mohammad  ↵
60.0      ↵
Please enter the ID, name, and score of student 1:
433000222 ↵
Ahmad     ↵
100.0          ↵
Please enter the ID, name, and score of student 2:
433000333 ↵
Khalid    ↵
50.0      ↵
The array after removing the Student at index 1:
433000111, Mohammed, 60.0
433000333, Khalid, 50.0
```

Sample run using `removeAndShiftStudents`

```
Please enter the ID, name, and score of student 0:
433000111 ↵
Mohammad  ↵
60.0      ↵
Please enter the ID, name, and score of student 1:
433000222 ↵
Ahmad     ↵
100.0          ↵
Please enter the ID, name, and score of student 2:
433000333 ↵
Khalid    ↵
50.0      ↵
```

```
The array after removing the Student at index 0:
433000222, Ahmad, 100.0
433000333, Khalid, 50.0
433000333, Khalid, 50.0
```

Complete following pseudo code

```
class CourseManager8 {
        /* Declare the class data members as shown in the UML or
    copy from CourseManager6 and paste here*/

    public CourseManager8() {
      /* copy from CourseManager6 and paste here* */
         }

     public /* method type */ addStudent(/* students' object */)
{

             /* copy from CourseManager6 and paste here*   */
    }


    public /* method type */ findStudentName(/* parameters list
*/) {

            /* copy from CourseManager6 and paste here */
    }

    public /* method type */ findAverageScore(/* parameters list
*/) {
            /* copy from CourseManager7 and paste here */
     }

    public /* method type */ findMaxScoreIndex(/* parameters list
*/) {

            /* copy from CourseManager7 and paste here
    */
}

    public /* method type */ displayStudent(/* parameters list
*/) {

            /* print the id, name , and scores of the index i
passed to the method */
    }

    public /* method type */ getNStudents(/* parameters list */)
{

                /* return nStudents */
```

```
    }

    public /* method type */ removeStudent(/* parameters list */)
{

                /* set the value of the object at index i to
null */
    }

public /* method type */ removeAndShiftStudent(/* parameters list
*/) {

                /* set the value of the object at index i++ to
i
     Continue doing this for all the elements after  index i
till the end of the array*/
    }
}

import java.util.Scanner;
public class TestCourseManager7 {
    public static void main(String[] args) {
     Scanner kb = new Scanner(System.in);

       /* creat CourseManager8 object named c1 */

        /* use for loop to do the following 4 times:

        /* 1- ask the user to enter student information ID, name,
and score */
        //System.out.println("Please enter the ID, name, and
score of a student: ");

        /* 2- use the scanner to get the id, name , and score */

    /* 3- create a student with the new information. */

        /* 4- add student using the method "addStudent" */

        /* after the loop remove the student at index I using
removeStudentt() */
        /* print all the students information */

    /* rerun the program and use removeAndShifStudents() instead
to remove the student at index 0 */
        /* print all the students information */

       }}
```