King Saud University

College of Computer and Information Sciences
Computer Science Department

Duration :	**3 hours**
Course Code:	CSC 111
Course Title:	Introduction to Programming
Semester:	Fall 2014
Exercises Cover Sheet:	**Final Exam**

Student Name:	
Student ID:	
Student Section No.	

Tick the Relevant	Computer Science B.Sc. Program ABET Student Outcomes	Question No. Relevant Is Hyperlinked	Covering %
√	a) Apply knowledge of computing and mathematics appropriate to the discipline;	2	30
	b) Analyze a problem, and identify and define the computing requirements appropriate to its solution		
√	c) Design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;	1,3,4	70
	d) Function effectively on teams to accomplish a common goal;		
	e) Understanding of professional, ethical, legal, security, and social issues and responsibilities;		
	f) Communicate effectively with a range of audiences;		
	g) Analyze the local and global impact of computing on individuals, organizations and society;		
	h) Recognition of the need for, and an ability to engage in, continuing professional development;		
√	i) Use current techniques, skills, and tools necessary for computing practices.		
	j) Apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;		
	k) Apply design and development principles in the construction of software systems of varying complexity;		

Question 1: (8 Marks)

a- Draw the UML class diagram of the following Java code:

```
public class NamesList {
   private String name[];
   private int code[];
   private int number;

   public NamesList(int z){
         name = new String[z];
         code = new int[z];
         number = 0;
   }
   public boolean AddNewRec(int ident, String mn){
         if (number<code.length){
               code[number]=ident;
               name[number] = new String(mn);
               number++;
               return true;
         }
         else
               return false;
   }
   public void display(){
         int i;
         for (i=0;i<number;i++){
               System.out.println(code[i]+" --- "+name[i]);
         }
   }
   public String display(int location){
         if (location>=0 && location<number)
               return name[location];
         else
               return "not present";
   }
   public void deleteByLastIndex(int location){
         if (location>=0 && location<number){
               code[location]= code[number-1];
               name[location]=name[number-1];
               number--;
         }
   }
   public int deleteByShiftingData(int location){
         if (location>=0 && location<number){
               for (int i= location; i<number-1; i++){
                     code[i]= code[i+1];
                     name[i]=name[i+1];
               }
               number--;
         }
         return number;
   }
}
```

Answer:

NamesList
- Name : string[]
- Code : integer[]
- Number : int
+ NamesList(integer)
+ AddNewRec(integer, string) : bool
+ display()
+ display(integer): string
+ deleteByLastIndex(integer)
+ deleteByShiftingData(int)

.....	0.50
.....	0.50
.....	0.50
.....	0.50
.....	1.00
.....	1.00
.....	1.00
.....	1.00
.....	1.00
.....	1.00

Question 2: (12 Marks)

I) Answer the following with True/False:

Question	Answer (True/False)
a. The output of the following code segment is "26": `int someInt=26; System.out.println("someInt");`	False .. 1
b. To declare a boolean variable b in Java we write: `boolean b;`	True ... 1
c. In java, a variable x is divisible by 5 if the expression `(x%5==0)` is evaluated true.	True ... 1
d. Using a *switch*-statement only (without using if), we can know whether an integer x is equal to 0 or not.	True ... 1
e. Compiling a program successfully guarantees that it executes correctly.	False .. 1

f. The following two lines of code do the same thing:
`for(i=1;i<5;i++) z = z+i;`
AND
`i=1; do { z += i; i = i + 1; } while (i<5);`
	True ... 1

g. The following statement is valid in Java `int x=1; double y=2.0; x = x+y;`	False .. 1
h. The expression `(x = 0)` is used in an *if*-statement to know whether x is equal to 0.	False .. 1

II) Multiple choice: Circle only one choice in each question:

a. After executing the following code segment:

```
int result=1; for(i=1;i<=4;i++) result = result*i;
```

the variable result will contain the value:

a) 20
b) 24 ..... 1.00
c) 30
d) 10

b. Given two variables x and y of type int, where we assume that x=3, and y=4, we calculate z as follows:

```
int z=0; for(i=0;i<x;i++) z += y;
```

what will the value of z be?

a) 0
b) 3
c) 8
d) 12 ..... 1.00
e) Null

c. What will the value of v be after executing the following code segment?

Suppose that the value stored in 'a' are

5	8	3	0

```
v=a[0];
for(i=1;i<a.length;i++) {if (a[i]>v) v=a[i]};
```

a) 0
b) 8 ..... 1.00
c) 4
d) 5
e) Null

d. Given a Boolean variable w, which of the following is a valid assignment statement?

a) w = false;
b) w = w && w;
c) w = 5 > n;
d) a and c above
e) a, b, and c above 1.00

e. What is the output of the following code segment:

```
System.out.print(10/3);
```

a) 3 ..... 1.00
b) 3.3
c) 3.33333333
d) none of the above

Question 3 : (10 Marks)

Implement in Java the following `CourseSection` class diagram:

CourseSection
- courseName : String - nameOfStudent : String[] - studentID : integer[] - GPA : double[] - nbStudents : integer
+ CourseSection(capacity : integer, cName: String) + addNewStudent(stName:String, stID:integer, stGPA:double):boolean + getStudentIndex(stId : integer) : integer + getnbStudents() : integer + deleteStudent(stId : integer) : boolean + displayStudentAtIndex(k : integer) : void + displayPassOrFailStudents (indicator : String) : void

Attributes :

- nameOfStudent : Array that stores names of students registered in the section.
- studentID : Array that stores IDs of students registered in the section.
- GPA : Array that stores gpa of each student registered in the section.
- nbStudents : Number of students registered in the section currently.
- courseName : Name of the course (e.g. Programming 1)

Methods :

a) `CourseSection(capacity : integer, cName: String)`: A constructor method which creates arrays with the given capacity (the maximum number of students that a section can have) and assigns cName to courseName and initializes the attributes nbStudents to zero.

b) `addNewStudent(stName:String, stID:integer, stGPA:double)` : Adds a new student in the section with the studentName as stName, studentID as stId and GPA as stGPA. This method returns true if the student is added successfully otherwise it returns false.

c) `getStudentIndex(stId : integer)`: returns index of the student whose ID is stId otherwise -1.

d) `getnbStudents()`: returns total number of student currently registered in the section.

e) `deleteStudent(stId : integer)`: deletes a student whose ID is stId. This method returns true on successful deletion otherwise it returns false.

f) `displayStudentAtIndex(k : integer)`: Shows data of the student whose is stored at position k.

g) `displayPassOrFailStudents(indicator : String) : void`: Shows information of all the students in the section whose indicator is according to the following table:

indicator	GPA
Pass	>=2.5
Fail	<2.5

If the indicator is equal to Pass then the method would display ONLY the students who passed. If the indicator is equal to Fail then the method would display ONLY the students who failed.

Answer:

```
class CourseSection {
  private String courseName; ..........................................      1.00
  private int nbStudents; .............................................0.25
  private String[] nameOfStudent; .....................................0.25
  private int[] studentID; ............................................0.25
  private double[] GPA; ...............................................0.25

  public CourseSection(int capacity, String cName) { ..................0.25 1.50
    courseName = cName; ...............................................0.25
    nameOfStudent = new String[capacity]; .............................0.25
    studentID = new int[capacity]; ....................................0.25
    GPA = new double[capacity]; .......................................0.25
    nbStudents = 0; ...................................................0.25
  }

  public boolean addNewStudent(String stName, int stID, double stGPA) { .0.25 2.00
    if (nbStudents == nameOfStudent.length) return false; .............0.25
    if (getStudentIndex(stID) != -1) return false; ....................0.25
    nameOfStudent[nbStudents] = stName; ...............................0.25
    studentID[nbStudents] = stID; .....................................0.25
    GPA[nbStudents] = stGPA; ..........................................0.25
    nbStudents++; .....................................................0.25
    return true; ......................................................0.25
  }

  public int getStudentIndex(int stID) { ..............................0.25 1.00
    for (int i=0; i < nbStudents; i++) ................................0.25
      if (studentID[i] == stID) return i; .............................0.25
    return -1;   ......................................................0.25
  }

  public int getNbStudents() { ........................................0.25 0.50
    return nbStudents; ................................................0.25
  }

  public boolean deleteStudent(int stID) { ............................0.25 2.00
    int indx = getStudentIndex(stID); .................................0.25
    if (indx == -1) return false; .....................................0.25
    nbStudents--; .....................................................0.25
    nameOfStudent[indx] = nameOfStudent[nbStudents]; ..................0.25
    studentID[indx] = studentID[nbStudents]; ..........................0.25
    GPA[indx] = GPA[nbStudents]; ......................................0.25
    return true; ......................................................0.25
  }

  public void displayStudentAtIndex(int k) { ..........................0.25 1.00
    if (k >= 0 && k < nbStudents) { ...................................0.25
      System.out.println( "Student's name:" + nameOfStudent[k] + "\n" +
                          "Student's ID:" + studentID[k] + "\n" +
                          "Student's GPA:" + GPA[k] ); ................0.50
    }
  }

  public void displayPassOrFailStudents(String indicator) { ...........0.25 1.00
    for (int i=0; i<nbStudents; i++) ..................................0.25
      if ( (indicator.equals("Pass") && GPA[i] >= 2.5) ||
           (indicator.equals("Fail") && GPA[i] < 2.5) ) ...............0.25
        displayStudentAtIndex(i); .....................................0.25
  }
}
```

Question 4 : (10 Marks)

Using class `CourseSection`, write a Java class `CourseSection`, which contains a main program that performs following operations:

a) Create CourseSection object with capacity of 45 student and course name "Programming 1".
b) Add following students in the section.

Student Name	ID	GPA
Ali	42910011	2.6
Fahad	43121476	2.1
Majid	43395138	3.5

c) (we suppose that we inserted several students) Delete student whose ID is 43395138 from the section.
d) Display information about the student whose ID is 43122345.
e) Display all information about all students.
f) Display all information about all students whose grade is "Pass".

```
public class CourseTest{
  public static void main(String args[]){

    //create CourseSection object here. (part a)
    CourseSection cs = new CourseSection(45, "Programming 1"); ..... 1.25

    //add 3 students as given in the table above (part b).
    cs.addNewStudent("Ali", 42910011, 2.6);  ..... 0.75
    cs.addNewStudent("Fahad", 43121476, 2.1);  ..... 0.75
    cs.addNewStudent("Majid", 43395138, 3.5);  ..... 0.75

    //assume now there are many registered students in the section.
    //delete student whose ID is 43395138 from the section.  (part c)
    cs.deleteStudent(43395138);  ..... 1.00

    // display information about the student whose ID is 43122345. (part d)
    int i = cs.getStudentIndex(43122345);  ..... 1.00
    cs.displayStudentAtIndex(i);  ..... 1.00

    //display all information about all students. (part e)
    int nb = cs.getNbStudents(); ..... 0.75
    for(i = 0; i<nb; i++)  ..... 0.75
      cs.displayStudentAtIndex(i);  ..... 0.75

    //display all info about all students whose grade is "Pass". (part f)
    cs.displayPassOrFailStudents("Pass");  ..... 1.25

  }
}
```

Result

Question No.	Relevant Student Outcome	SO is Covered by %	Full Mark	Student Mark	Assessor's Feedback
1	a	20	8		
2	c	30	12		
3	c	25	10		
4	c	25	10		
Totals		100%	40		

I certify that the work contained within this assignment is all my own work and referenced where required. **Student Signature:** **Date:**		**Feedback Received:** **Student Signature:** **Date:**