KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
COMPUTER SCIENCE DEPARTMENT

CSC 113: Computer Programming II	**Final Exam** **(Duration: 3 Hours)**	**2nd Semester 1438-1439**

Student Name (Arabic)	Student ID	Section Number	Serial Number

Question#1: Multiple Choice Questions *(10 pts.)*

For each statement there is a list of options. Choose the option that would be the valid one.

1. What will be the output of the following piece of code:

```
class Person{
  public void talk() {}}
public class Test{
  public static void main(String args[]){
    try{
      Person p = null;
      try{
        p.talk();
      }
      catch(Exception e){
        System.out.println("There is an Exception. "); }
    }
    catch(NullPointerException e){
      System.out.println("There is a NullPointerException. "); }
    System.out.print("Everything went fine. ");}}
```

A. Compilation Error.
B. There is a NullPointerException.
   Everything went fine.
C. Everything went fine.
D. There is an Exception.
   Everything went fine.

2. When using a PrintWriter it is possible to do output with which method?

A. write()
B. writeln()
C. println()
D. out()

3. When writing to a file with a FileOutputStream, which type is valid?

A. int
B. char
C. byte
D. All of the above

4.What will be the output of the following piece of code-if any-:

```
abstract class C1{
        public C1(){
                System.out.print(1); } }
class C2 extends C1{
        public C2(){
                System.out.print(2); } }
class C3 extends C2{
        public C3(){
               super();
                System.out.println(3); } }
public class Test{
        public static void main(String[] a){
                C1 obj=new C3(); } }
```

A. Compilation Error
B. 3
C. 23
D. 123

5. Suppose L refers to a linked list. Which of the following boolean expressions is true when L is a list with one node?

A. (L.getHead() == L.getTail())

B. (L.getTail() == null)

C. (L.getHead().next == L.getTail())

D. (L.getHead().next.next == null)

6. What methods should be used when implementing Queue using a linked list?

A. insertAtFront() and removeFromFront()

B. insertAtBack() and removeFromFront()

C. insertAtBack() and removeFromBack()

D. None of the above

7. **What will be the output of the following piece of code-if any-:**

```
public class testGeneric {
   public static void main (String args []) {
      LinkedList<Integer> iList=new LinkedList<Integer>();
      iList.insertAtBack(new Integer(-5));
      iList.insertAtFront(new Integer(20));
      iList.insertAtFront(new Integer(-4));
      iList.insertAtBack(new Double (3.0));
      iList.print();}}
```

A. -4 20 -5 3.0

B. -5 20 -4 3.0

C. 3.0 -4 20 -5

D. Compilation Error

8. If a class that implements an interface does not implement all the methods of the interface, then the class must be a/an……………………. class.

A. abstract

B. final

C. static

D. supe

9. **What will be the output of the following piece of code if any:**

```
public class MyClass {
    public static int Secret (int x){
        int sum=0;
        System.out.println(x+" ");
if (x==4) return 1;
        if(x < 2)
          sum+=Secret(++x);
         else
           sum+=Secret(x++);
        return sum;  }
    public static void main(String args[]) {
        System.out.println(Secret(1));}
}
```

A. Compilation error

B. 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

C. 1 2 2   2 1

D. Infinite output

10. What will be the output of the following piece of code if any:

```
public class MyClass {
    public static int Secret (int x)
    {
        if(x == 0)
          return 0;
        return ((x%10)+Secret(x/10));
    }
    public static void main(String args[]) {
        System.out.println(Secret(12)); }}
```

A. Compilation error

B. 12 3 1 0

C. 3

D. Infinite output

11. **The following code will compile successfully**

```
import java.io.*;
public class finalExam {
 public static void main(String[] args) {
    File f = new File ("output.txt");
    FileOutputStream out = new FileOutputStream(f);
     byte b [] = {11,21,3,40,5};
       if(f.exists()){
         out.write(b);
         out.close();
       }
       else{
     System.out.println("file doesn't exists ");
System.exit(1);
       }
 }
}
```

A. True
B. False

12. The following code will compile successfully

```
public abstract class xy
{
public abstract  void  m1 (int x, int y) { }
}
```

A. True
B. False

13. The following code segment will create new logical file name f linked to a.txt?

```
          try{
               File f = new File("a.txt");
          }
          catch(Exception e){}
          catch(IOException io){}
```

A. True
B. False

14. The following code segment will print 5^2=25

```
public class Test{
  public static void main(String args[])
    {
     System.out.println ("5^2 ="+recMethod(5,2));
     }

     public static int recMethod(int b, int p )
     {
           return b*recMethod(b, p-1 );
      }
}
```

A. True
B. False

15. The following statement will print the last node in the linked list

```
          System.out.println (tail.getNext());
```

A. True
B. False

16. The following code will print 0   1

```
class A {
        public int i;
        private int j;}
class B extends A {
          public int i;
        void display()
        {
            super.j =i + 1;
            System.out.println(super.i + " " + super.j);
        }}
 public class inheritance
    {
        public static void main(String args[])
        {
            B obj = new B();
            obj.i=1;
             obj.display();
        }}
```

A. True

B. False

17. The following boolean expressions is true when L is an Empty list?

```
                L.getHead() == null
```

A. True

B. False

18. The following code will compile successfully

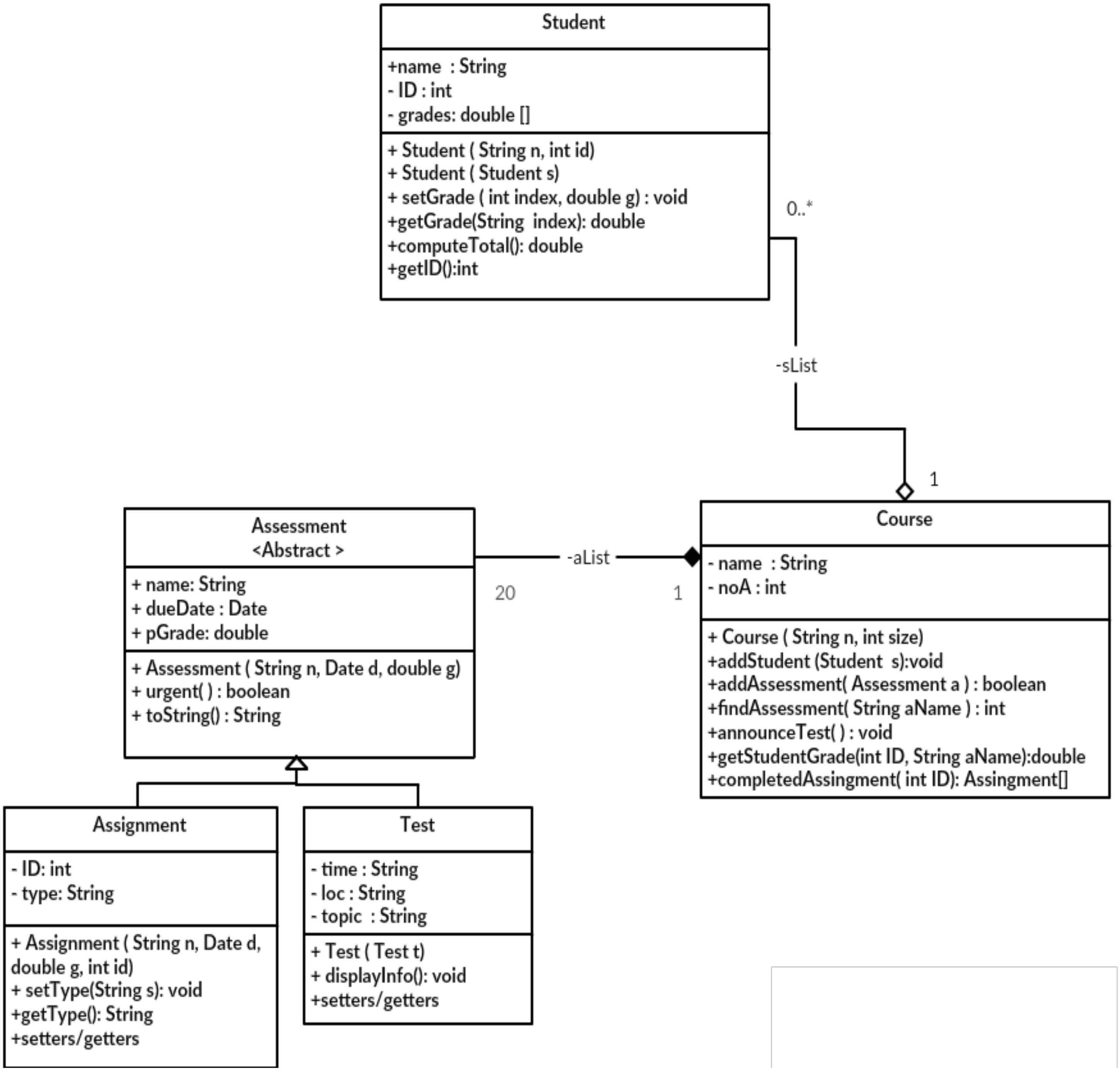
```
public interface A {
   public void m1();}
public interface B extends A {
  public  void m4();}
public class E implements B {
\\…
public void m1(){}
public void m4(){}}
public class test
    {
        public static void main(String args[])
        {
            A obj = new E();
             obj.m4();
        }
    }
```

A. True

B. False

Question#2: *(11 pts.)*

Consider the following UML and corresponding classes description



Class Assessment

String name	Name of the assessment.
Date dueDate	Due date of the assessment.
double pGrade	Possible grade for this assessment.
Assessment(String n, Date d, Double g)	Constructor for initializing assessment attributes.
urgent (): boolean	Return true if this assessment is urgent by comparing the due date with toady’s date.
toString():String	Return a formatted string of assessment’s information.

Class Assignment

int ID	Assignment number.
String type	Type of the assignment (*Homework*, *Lab*, *Project*)
Assignment(String n, Date d, Double g,int id)	Constructor for initializing assignment’s attributes from received parameters. **Set type to a default value “*Lab*”.**

Class Test

String time	Start time of the test.
String loc	Location of the test.
String topic	Chapter name covered by the test.
Test (Test t)	Copy Constructor.
displayInfo():void	Print Test information, **including *Name, Date, Time, Location and Topic.***

Class Student

String name	Student name.
int ID	Student ID.
double grades[]	An array contains 20 grades for the student, each element contains grade for specific assessment in the course stored in alist. The grade in index 0 in grades is for the assessment in index 0 in alist and so on.
Student (Student s)	Copy Constructor
Student(String n, int id)	Constructor for initializing Student’s attributes, initialize **grades** elements with -1
setGrades (int index, double g):void	Set grade g at the specified position index in the array **grades.** grades[index]=g;
getGrades (int index):double	Returns the grade at the specified position in **grades**. return grades[index];
computeTotal(): double	Return total grades for students, this method adds grades that have been set.

Assuming all above classes are implemented and given the following description of class course

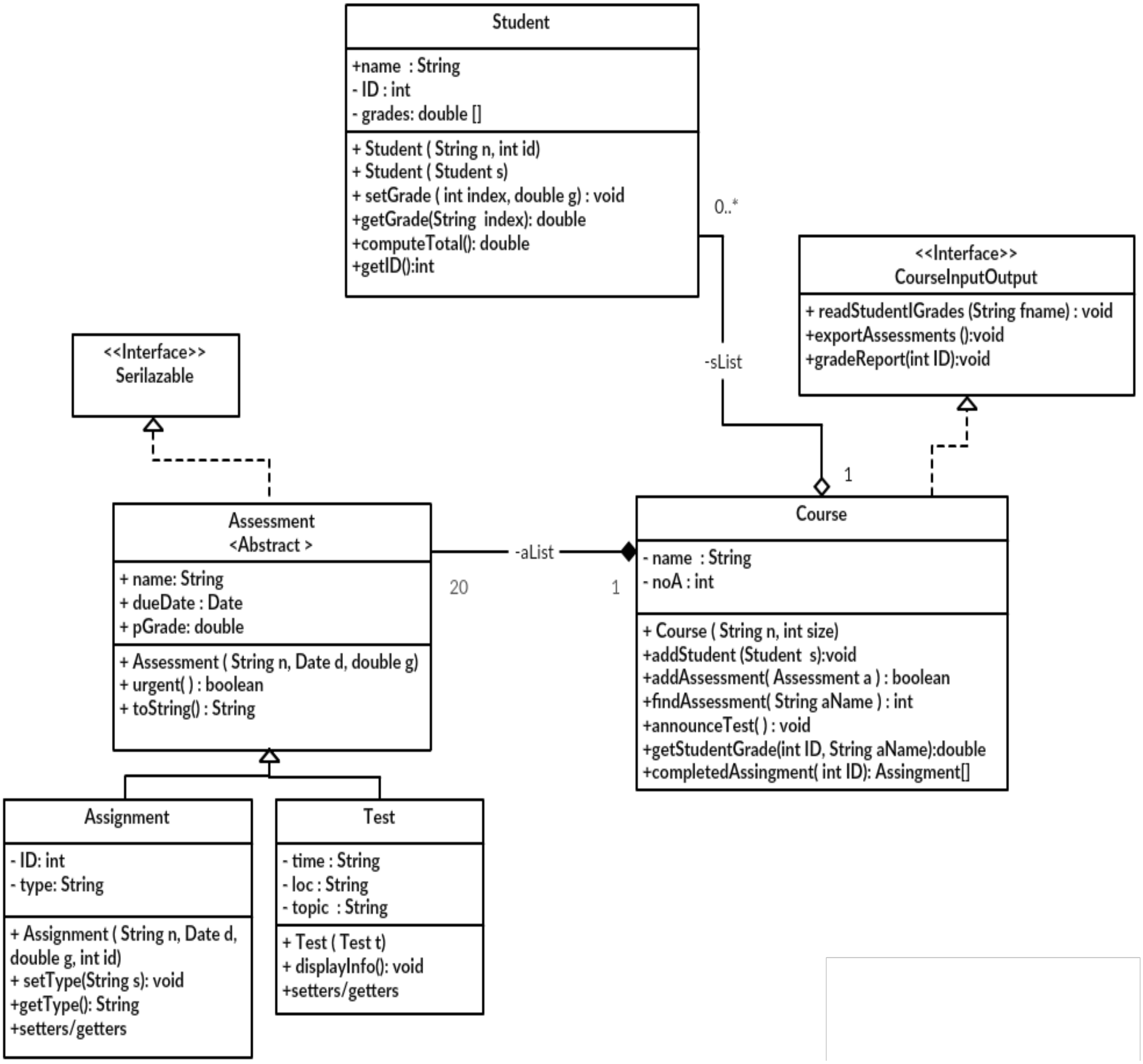
Class Course

String name	Course Name.
int noA	Number of current assessments in **aList**
Course (String n, int size)	Constructor for initializing course attributes
addAssessment(Assessment a):boolean	Assessment a: assessment to be added Add assessment **a** to ***aList*** if possible and return true**.**
AddStudent(Student s):void	Student s: student to be added Add student ***s*** to the course **at first empty** position in **sList.** If and only if the student **s** is **not exists** in **sList**. Note: Each student has **unique ID**
announceTest():void	Print information for all upcoming **urgent** Tests in **aList.**
getStudentGrade(int ID, String aName): double	Return grade for student with received ID for assessment has the name **aName.**
findAssessment(String aName): int	Return index for assessment in **aList** array has name **aName**, -1 if not found
completedAssignment(int ID): Assignment[]	Return an array of completed assignments only for student with the received ID. Completed assignment for student is the assignment with grade not equal to -1

1. In class Course, write or complete the Java implementation of the following methods:
   - *a.* addAssessment(Assessment a):boolean ***(4pts)***
   - *b.* addStudent(Student s):void ***(4pts)***
   - *c.* announceTest():void ***(3pts)***

Question#3:*(10 pts.)*

Consider the following updated UML from question 2:


Student
+name : String
- ID : int
- grades: double []
+ Student (String n, int id)
+ Student (Student s)
+ setGrade (int index, double g) : void
+getGrade(String index): double
+computeTotal(): double
+getID():int
0..*
-sList
1
<<Interface>>
CourseInputOutput
+ readStudentIGrades (String fname) : void
+exportAssessments ():void
+gradeReport(int ID):void
<<Interface>>
Serilazable
Assessment
<Abstract >
+ name: String
+ dueDate : Date
+ pGrade: double
+ Assessment (String n, Date d, double g)
+ urgent( ) : boolean
+ toString() : String
-aList
20
1
Course
- name : String
- noA : int
+ Course (String n, int size)
+addStudent (Student s):void
+addAssessment(Assessment a) : boolean
+findAssessment(String aName) : int
+announceTest( ) : void
+getStudentGrade(int ID, String aName):double
+completedAssingment(int ID): Assingment[]
Assignment
- ID: int
- type: String
+ Assignment (String n, Date d, double g, int id)
+ setType(String s): void
+getType(): String
+setters/getters
Test
- time : String
- loc : String
- topic : String
+ Test (Test t)
+ displayInfo(): void
+setters/getters

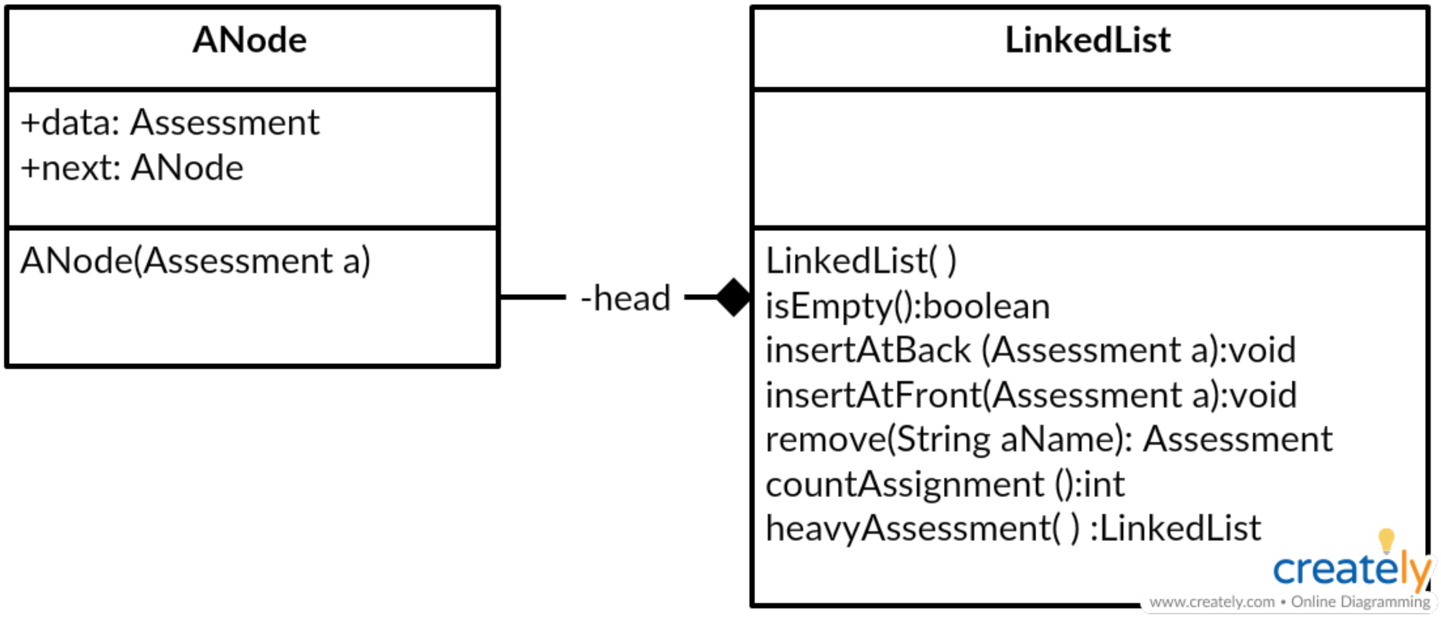
Class Course

readStudentsGrades (String fName):void	String fName : name of text file contains students grades in different assessments. This method reads *student id, assessment name*, and *grade* from a **text file** named fname. The method should **set the grade for the student** in **sList** for that assessment. If the assessment is not found in **aList** then continue reading the next line. If student is not found in **sList** print “Student with ID *** not found” and continue reading the next line. This method should handle all the exceptions that may occur during the reading process by printing appropriate message. **Hint: Use the method setGrades (int index, double g)** **File Format** (student ID:int)~(assessment name: String)~(grade: double) where~represents space **Example:** 437105 ~HW1 ~ 1.0 437107~Quiz1~2.5 436109 ~Sheet4 ~2.0 …………….
exportAssessments():void	This method writes all assessments in **aList** at once into an object file named “assessments.data”. This method should handle all the exceptions that may occur during the writing process by printing appropriate message.
gradeReport(int ID) :void	int ID: student ID This method should write into a text file named **ID+”Grades.txt**”, the grades for all completed assignments for the student who has the received ID. **This method should handle all the exceptions that may occur during the writing process.** **Hint: Use the method getGrades (double g)** Note: Assume ID is valid id

1. In class Course, write or complete the Java implementation of the following methods:
   a. readStudentsgrades (String fName):void ***(4pts)***
   b. exportAssessments():void ***(4pts)***
   c. gradeReport(int ID):void ***(2pts)***

Question#4: ***(9 pts.)***

Using Class Assessment from previous question and the following UML


ANode
+data: Assessment
+next: ANode
ANode(Assessment a)
-head
LinkedList
LinkedList( )
isEmpty():boolean
insertAtBack (Assessment a):void
insertAtFront(Assessment a):void
remove(String aName): Assessment
countAssignment ():int
heavyAssessment( ) :LinkedList
creately
www.creately.com • Online Diagramming

Class LinkeList

remove(String aName): Assesment	This method removes from linkedList the assessment with the received name **aName** and return the removed assessment, return **null** if the assessment is not found. **Note: consider all possible cases.**
countAssignment():int	This method should return the number of assessment of type ***assignment*** in the linked list.
heavyAssessment(): Linked List	This method should return a new list of all assessment that have the possible grades greater than 15.

1. Write the Java implementation of the following methods:
   a. remove (String aName) : Assessment ***(4pts)***
   b. countAssignment():int ***(3pts)***
   c. heavyAssessment(): Linked List ***(2pts)***

KING SAUD UNIVERSITY COLLEGE OF COMPUTER AND INFORMATION SCIENCES COMPUTER SCIENCE DEPARTMENT		
CSC 113: Computer Programming II	**Final Exam (Duration: 3 Hours)**	**2nd Semester 1438-1439**

Answer Sheet

Student Name (Arabic)	Student ID	Section Number	Serial Number

Question#1: Multiple Choice Questions ***(10 pts.)***

For each statement there is a list of options. Choose the option that would be the valid one.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
D **0.5**	**C** **0.5**	**C** **0.5**	**D** **0.5**	**A** **0.5**	**B** **0.5**	**D** **0.5**	**A** **0.5**	**D** **1.5**
(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)
C **1.5**	**B** **0.5**	**B** **0.5**	**B** **0.5**	**B** **0.5**	**B** **0.5**	**B** **0.5**	**A** **0.5**	**B** **0.5**

Commented [MP1]: Choices are updated to be Consistent with previous points

Question#2:

a. Complete the following implementation for method addAssessment

```
      public boolean addAssessment(Assessment a) {
   // add Assessment to this course
             if (noA==aList.length)   // 0.5
                    return false;
//add test to alist
             if(a instanceof Test) {

                    aList[noA]=new Test((Test)a);
              //1.5 (aList[noA]0.5+new Test 0.5+(Test)a 0.5)
             }
// add assignment to aList
             else  if(a instanceof Assignment) {
aList[noA]=new Assignment(a.name,a.dueDate,a.pGrade,((Assignment)a).getID());
//1.5 (new Assignment 0.5+ a.name,a.dueDate,a.pGrade 0.5+
((Assignment)a).getID()) 0.5)
((Assignment)aList[noA]).setType(((Assignment)a.getType()))
//1  ((Assignment)aList[noA]).setType 0.5 + (((Assignment)a.getType()))
+0.5
             }
             noA++;
             return true;  }
```

b. Implemnt method method addStudent

```
      public void AddStudent(Student s){
          int index=-1;   // variable to check if student s exists or not using
index or/and flag
     //0.5 for variable declaration
      for (int i=0; i<sList.length;i++)
//.5 for correct for loop condition
           if (sList[i]==null)
           {
                index=-1;
                break;
           }
// 1 for find first empty position (sList[i]==null 0.5 +index=-1 0.5);
)
     else
      if(slist[i].getID()==s.getID())
          return;
//1 no add if student exist
if (index !=-1)
         sList[index]=s;
// 1  adding s if and only if there is empty place
}
```

c. Implemnt method announceTest

```
     public void announceTest(){

     for (int i=0;i<noA;i++)
  //0.5 for correct for loop condition
      if(aList[noA] instanceof Test && aList[noA].urgent())
// 1.5 aList[noA] instanceof Test 0.75+ aList[noA].urgent() 0.75
                    ((Test)aList[noA]).displayInfo();
// 1  ((Test)aList[noA]) 0.5 for casting + displayInfo()0.5;
```

Question#3:

a. Complete the following implementation for method readStudentsgrades

```
public void readStudentsGrades (String fName){
Scanner input=null;
try {
     Scanner input =new Scanner (new File(fName));
     int id=-1;
     String aName=null;
     double g=-1;
          while(input.hasNext())
   //0.5 for correct condition
     {
                id=  input.nextInt();
                aName= input.next();
                g= input.nextDouble();
 //0.75 (0.25 for each correct read )
          int aIndex= findAssessment(aName);
          if (aIndex ==-1)
              continue;
          int sIndex=-1; // index for student
          // find student index in sList
             for (int i=0;i<sList.length;i++)  //0.5
                  if(slist[i].getID()==id){  //0.5
                index=i;    //0.5
                break;}
          // setGrade g for student
          try {
            sList[index].setGrade(aIndex,g);
  // 1 (sList[index] 0.5+ setGrade(aIndex,g) 0.5)
            catch (IndexOutOfBoundsException e){
      System.out.println("Student with ID"+ id+" not found");
//0.25
}

     }//end while
   input.close();  //0.5
     }//end try
     //exception during processing file
     catch(IOException e)   //0.5
     {
          System.out.println("Exception "+e);
     }
}
```

b. Implement method ExportAssessments()

```
public void ExportAssessments(){

   try {
          //create appropriate stream
       File f=new File ("Grades.data");
          FileOutputStream  fos=new FileOutputStream(f);
          ObjectOutputStream oos=new DataOutputStream(fos);  1.5 ( .5 for
each new )
//write assessments in aList at once
 oos.writeObject(alist);  //1
  oos.close();    //0.5
}//exception during processing file
catch (IOException e) { //0.5
System.out.println("Exception while processing file"+e.getmessage()); //0.5
}
}
```

c. **The following implementation is incorrect for method gradeReport()? Explain why?**

```
public void gradeReport(int ID){
                    1. Assignment [] cList= completedTasks(ID);
                    2. File f=new File (ID+"Grades.data");
                    3. FileOutputStream  fos=new FileOutputStream(f);
                    4. PrintWriter pr=new PrintWriter(fos);
                    5. for (int j=0; j<cList.length;j++) {
                    6.        double g=getStudentGrades(ID,cList[j].getName());
                    7.        pr.print(cList[j]);
                    8.        pr.println("Student grade is" +g);}
                    9. }//end method
```

Line	Reason /Correction	
2+8	missing try catch	//1
8	pr.close() is missing	//1

Question#4:

a. **Implement method remove**

```
    public Assessment remove(String aName){
        if (isEmpty())0.25
            return null; 0.25
     // 0.5 for empty list case
     ANode current = head;
        ANode pre = null;
 while(current!=null && !current.data.name.equals(aName)) 0.5->0.25 each
            {
                pre = current; 0.5
              current=current.next; 0.5
            }
// 1.5 for finding node and its previous node
          Assessment e=current.data;
        if (current == head) 0.25
            head=null; 0.25
//0.5 for head case
```

```
    else
      pre.next=current.next;
//1 for remove node by linking prev to current.next
    return e;
//.5 for return e or current.da
```

b. Implement method countAssignment

```
public int countAssignment(){

            if (isEmpty())
                return 0;

             int count=0;  //0.25
               ANode current = head;  // 0.5
               while (current != null)  //0.5
               {
     if (current.data instanceof Assignment) //0.5
                   count++;     //0.5
                 current = current.next;  //0.5
               }
               return count; //0.25
}
```

d. The following implementation is incorrect for method `heavyAssessment()`**? Explain why?**

```
public LinkedList heavyAssessment(){
   1. LinkedList hList=new LinkedList();
   2. ANode current = head;
   3. while (current.next!= null){
   4. if (current.data.pGrade > 15)
   5. hList.insertAtBack(current);
   6. current = current.next;}
   7. return hList;}
```

Line	Reason /Correction	
3	current!= null	//1
5	hList.insertAtBack(current.data);	//1