Started on	Sunday, 14 January 2024, 8:00 PM
State	Finished
Completed on	Sunday, 14 January 2024, 9:00 PM
Time taken	1 hour
Grade	**11.00** out of 15.00 (**73.33**%)

Question **1**

Complete

Mark 3.00 out of 3.00

Consider the following classes, which of the statements bellow are true?

```
public class Student {
     private int id;
     private BirthDate birthDate;
     public Student(int ssn, int year, int month, int day) {
          id = ssn;
          birthDate = new BirthDate(year, month, day);
      }

     public int getId() {
          return id;
     }

     public BirthDate getBirthDate() {
         return birthDate;
     }
}

/////Birth Date Class///////

public class BirthDate {
     private int year;
     private int month;
     private int day;

     public BirthDate(int newYear, int newMonth, int newDay) {
          year = newYear;
         month = newMonth;
         day = newDay;
     }

    public int getYear() {
      return this.year;
    }

    public int getMonth() {
      return this.month;
    }

    public int getDay() {
      return this.day;
    }
}
```

☑ a. The class 'Student' is aggregating and the class 'BirthDate' is aggregated

☐ b. The class 'BirthDate' is not immutable

☐ c. The class 'BirthDate' is aggregating and the class 'Student' is aggregated

☑ d. The class 'Student' is immutable

Your answer is correct.

Question **2**

Complete

Mark 3.00 out of 3.00

Consider the following three classes: C1, C2 and C3

```
package p1;
class C1{
      private int x;
      int y;
      public int z;
// constructor and methods are omitted
}
/// class C2//
package p1;
public class C2{
      public void m(){
       C1 c1 = new C1();
       c1.x =10;
       c2.y = 5;
       c3.z=  2;

       }
}
///class C3//
package p2;
public class C3{

      public void m(){
       C1 c1 = new C1();
       c1.x =10;
       c2.y = 5;
       c3.z=  2;
     }
}
```

which of the following statements are correct?

□ a. C3 can access C1.z

□ b. C2 cannot create instances from C1

☑ c. C2 cannot access C1.x

□ d. C2 cannot access C1.y

☑ e. C3 cannot create instances from C1

Your answer is correct.

Question **3**

Complete

Mark 3.00 out of 3.00

Using Wrapper class Integer, write a code that reads two integer values, saves in them Integer instances then perform multiplication between the Integer instances, and finally, print the result

(NOTE: no need to write a class or main method, just a code segment)
(NOTE 2: no need to read the data from user)

```
import java.util.*;
public class Test{
          public static void main(String[] args){
                    Scanner sc = new Scanner(System.in);
                     System.out.println("Please Enter the first integer);
                    int x = sc.nextInt();
                    System.out.println("Please Enter the second integer);
                    int y = sc.nextInt();
                     Integer obj1 = x ;
                    Integer obj2 = y ;
                    Integer res = obj1 * obj2 ;
                  System.out.println("The result = " + res);
                  }
  }
```

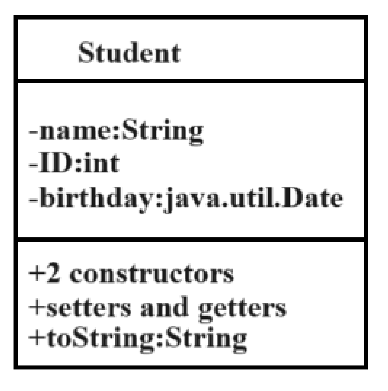
Comment:

Question **4**

Complete

Mark 2.00 out of 6.00

consider the following class UML:



write a *Driver* class that does the following:
- creates an array of *Student* instances with size decided by user
- ask user to fill the information for each instance in the array
- then send the array to a method (method in Driver class) called *sort(Student[] students),* the method sorts the array according to *Student.ID* in ascending order.
- finally, print the content of the array from the *main* method in *Driver* class after sorting it (hint: use the *toString()* method)

NOTE: you dont need to write the code for the Student class.

```
import java.util.* ;
public class Driver {
          public static void main(String[] args){
         Scanner sc = new Scanner(System.in) ;
         System.out.println("Please Enter the number of Students") ;
         int numOfStd = sc.nextInt() ;
         Student [] std = new Student[numOfStd] ;
         for(int i = 0 ; i < std.length ; i++ ) {
                     System.out.println("Please enter the name of student " + i ) ;
                     std[i].name = sc.next();
                     System.out.println("Please enter the ID for student " + i ) ;
                     std[i].ID = sc.nextInt();
                     System.out.println("Please enter the birthday for student " + i ) ;
                     std[i].birthday = sc.next();
                     }
        sort(std);


         }
       sort(Student [] x){
         String b;
        for(int j = 0; j<x.length ; j++) {
        b.contract([x[j]);
```

```
      }
     sort
      }


}
```

Comment: