

Name		QUID	
------	--	------	--

Quiz instructions: Answer all questions.

Q.1 [100 POINTS] [Students' average needed time to solve and review is 15 MINUTES]

- a. [40 POINTS] Code the enumeration **Dept** having the integer private attributes **capacity** and **count=0**. This enumeration declares three **Dept** objects each of a specified **capacity**. These objects are **CSE** with **capacity 300**, **EEE** with **capacity 100**, and **MIE** with **capacity 200**. The enumeration also has the public methods **getCapacity**, **getCount**, and **incCount**. The method **incCount** increments **count** by one as long as it does not exceed the value of the **capacity** otherwise, it does nothing.

- b. [50 POINTS] Finish the coding of the class **Student** by coding the following:

```
public class Student {
    private String id;
    private int year;
    private Dept dept;
    private String name;
}
```

- i. [10 POINTS] Code the method **genId** that generates and returns a string value for the attribute **id**. This value is a formatted string having the value of the attribute **year** followed by the name of the attribute **dept** followed by a three-digit integer representing the value of the attribute **count** of the **dept** object. The method return **null** for a **null Dept** object.
 - ii. [10 POINTS] Code the method **setId** that receives a **Dept** object that is used to set the value of the attribute **dept** then increments the value of the attribute **count** of the **dept** only if the **count** of the **dept** is less than the **capacity** of it. Otherwise, the method does nothing except displaying the message **"Error: capacity is reached"**.
 - iii. [10 POINTS] Code a constructor that receives a **Dept** object and use it to set the value of the attribute **deptId** by calling **setDept** method then it sets the attribute **deptId** to the value returned from calling the function **genId**.
 - iv. [10 POINTS] Override the method **toString** to return a formatted string having the values of all attributes.
 - v. [10 POINTS] Code the class method **equals** that takes two **Student** objects and returns **true** if all of their corresponding attributes are equal otherwise, it returns **false**.
- c. [10 POINTS] Create and print **100** anonymous **Student** objects with **dept** being **CSE** inside the constructor of the class **StudentTester**.

```
package l03Assessment.quiz2;
public class StudentTester {
    public StudentTester () {
    }
    public static void main(String[] args) {
        new StudentTester ();
    }
}
```

Name		QUID	
------	--	------	--

```
//a.  
package l02Assessment.quiz2;  
public enum Dept{  
    CSE(300),EEE(100),MIE(200);  
    private int capacity;  
    private int count=0;  
    Dept(int capacity) {  
        this.capacity=capacity;  
    }  
    public void incCount() {  
        if(count<capacity)  
            ++count;  
    }  
    public int getCount() {  
        return count;  
    }  
    public int getCapacity() {  
        return capacity;  
    }  
}
```

```

//b
package l02Assessment.quiz2;
public class Student {
    private String id;
    private int year;
    private Dept dept;
    private String name;
    //b.i.
    private String genId() {
        if(dept==null)
            return null;
        String id="";
        id = String.format("%04d%s%03d",year,dept,dept.getCount());
        return id;
    }
    //b.ii.
    public void setDept(Department dept) {
        if(dept.getCount()<dept.getCapacity()) {
            this.dept = dept;
            dept.incCount();
        }
        else {
            System.out.println("Error: capacity is reached");
        }
    }
    //b.iii.
    public Student(Department dept) {
        setDept(dept);
        id = genId();
    }
    //b.iv.
    @Override
    public String toString() {
        return String.format("Student [id=%s, year=%s, dept=%s, name=%s]", id,
                                year, dept, name);
    }
    //b.v.
    public static boolean equals(Student s1, Student s2) {
        return s1.name.equals(s2.name)&& s1.id.equals(s2.id)&&
            s1.dept==s2.dept&& s1.year==s2.year;
    }
}

//c
package l02Assessment.quiz2;
public class StudentTester {
    public StudentTester() {
        //c.
        for(int i=0;i<100;i++)
            new Student(Department.CSE);
    }
    public static void main(String[] args) {
        new StudentTester();
    }
}

```