Fall - 2023

1.a)

```
public class Student {

    private String name;
    private double grade;

public void setName (String name){
    this.name = name;
  }

public String getName (){
   return name;
  }

public void setGrade (double grade) {
    this.grade = grade;
  }

public double getGrade (){
  return grade;
  }
}
```

Output:

Alice Wonderland
95.5

1. b)

①
```
class Instrument {
    String name;

Instrument (String name) {
    this.name = name;

void play () {
 System.out.println ("Playing the" + name);
 }
void tune () {
 System.out.println ("Tuning the" + name);

 }
}
```

② class Guitor extends Instrument {

Guitor() {
super("Instrument");
}

(Default constructor যখন নাম ছাড়া object create হবে, তখন parent class Instrument এ "Instrument পাঠাবে)

Guitor(String form) {
super(form);
}

(এই constructor এ যে নাম pass করবো (like ukulele) সেটা parent class -এর constructor এ যাবে এবং name, variable এ save হবে)

@Override
void play() {
System.out.println("Strumming the guitor.");
}

void tune(String tuning){
System.out.println("Tuning the guitor to"+ tuning);
}

}

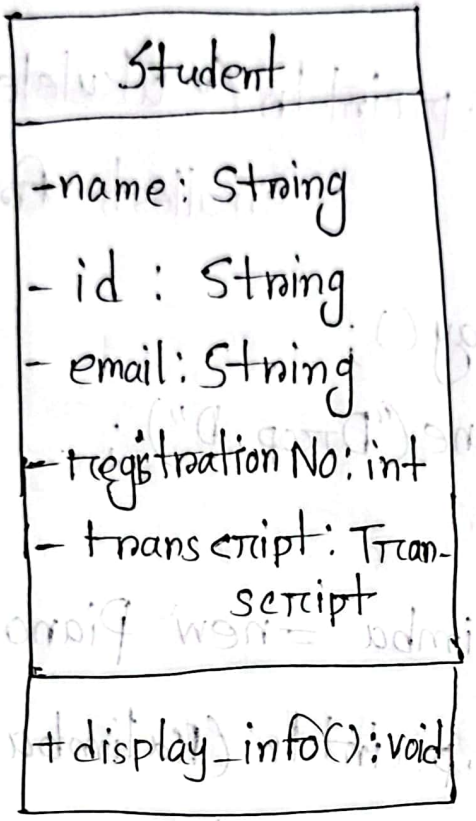
③
```
class Piano extends Instrument {

    piano(String fonm) {
        super(fonm);
    }

    tune(String tuning) {
        System.out.println("Tuning the piano to" + tuning);
    }
}
```

④
```
public class MusicStone {
    public static void main(String[] args) {

        Instrument instrumental = new Instrument("Instrument");

        instrumental.play();
        instrumental.tune();
```

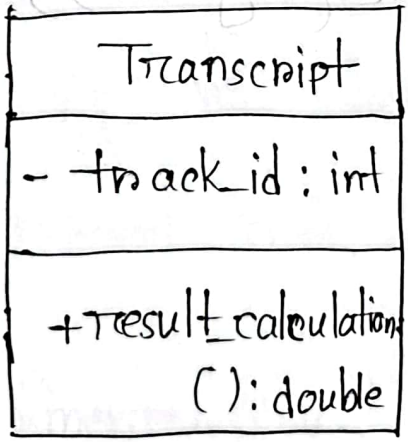
⑤
```
Guiter ukulele = new Guiten ("Ukulele");
System.out.println(ukulele.name + "is the
                    smallest from of Guiter");
ukulele.play();
ukulele.tune("Drop D");
```

⑥
```
Piano kalimba = new Piano ("Kalimba");
System.out.println(kalimba.name + "is also
                    Known as Pocket piano");
kalimba.tune("key 440");
}
}
```

1.c)


Student
+name: String
- id : String
- email: String
- registration No: int
- transcript: Transcript
+display_info(): void

has – a


Transcript
- track_id : int
+result_calculation(): double

2.a)

```
abstract class Flower {
    abstract void petal();
    void color() {
    System.out.println("The color jesmine is
                        a pale tint of yellow");
    }
}
class Jesmine extends Flower {
    void petal() {
    System.out.println("Jesmine petals are edible")
    System.out.println("Petals of Jesmine
                        moisturize our skin");
    }
}
public class Main {
    public static void main(String[] args) {
        Jesmin j = new Jesmin();
        j.petal();
        j.color();
} }
```

2.b) Same as - Spring 2024 (2.a)

2.c) Output:

OADFG4

2.d) Custom Exception for Voter Eligibility -

```
public class InvalidVoterException extends
                                    Exception {
public InvalidVoterException (String message){
    super(message);
    }
}
public class TestCustomException {
    static void validatedAge (int age)
    throws InvalidVoterException {
```

```
    if (age < 18) {
        throw new InvalidVoterException("Age is
                not enough to vote");
    }
    else {
        System.out.println("You are eligible to vote");
    }
}

public static void main(String[] args){
    try {
        validateAge(15);
    }
    catch (InvalidVoterException e){
        System.out.println("Found the exception");
        System.out.println("Exception occurred:"+e);
    }
}
```