Description:

This project consists of Student objects taking tests. The test consists of up to five java related questions which are given in a random order determining which version number the student gets. Then the students score gets calculated as a number out of 100 and a letter grade. There is a method which tells which student recieves the highest score on this test, and there is also a retake option for the students if they get below 80%. The best method is the method which determines whether or not the students get a pizza party, which they will get if the average is above 80.

Sample Output:

```
John's Test begins now:
True or False: Private variables can only be accessed in the same class
Answer:
true


True or False: OOP stand for Open Oven Pizza?
Answer:
false


Jack's Test begins now:
True or False: == compares the content in strings
Answer:
true


True or False: Double is a primitive data type
Answer:
true
```

```
Student's name: John, Date: 12/7/23, Score: 100
Student's name: Jack, Date: 12/7/23, Score: 50
The average is: 75.0
Highest score, Student's name: John, Date: 12/7/23, Score: 100
John grade is: A
Did not provide name, your grade is: F


No pizza party :(


You are not allowed to have a retake
75.0
Student's name: John, Date: 12/7/23, Score: 100
```

Code:

Main.java:

```
public class Main {
   public static void main(String[] args) {
        Test s1Test = new Test(2, "c"); //test for s1
        Test s2Test = new Test(2, "b"); //test for s2

        Student s1 = new Student("John","12/7/23" ,true, s1Test); //s1
object
        Student s2 = new Student("Jack", "12/7/23", false, s2Test); //s2
object

        s1.startTest(s1Test); //gives s1 the test
        s2.startTest(s2Test); //gives s2 the test



        System.out.println(s1); //prints the student name, date, and score
        System.out.println(s2);
        System.out.println("The average is: " + Test.calculateAverage());
//prints the test average out of two students

        if(s2.highestScore(s1,s2) == null) //if the students have the same
score
        {
            System.out.println("Students have the same score");
        }
        else {
            System.out.println("Highest score, " + s2.highestScore(s1,
s2)); //prints student with the highest score
        }
        System.out.println(s1Test.calculateGrade(s1)); //calculates letter
grade for s1
        System.out.println(s2Test.calculateGrade()); //s2 did not say who
they were so they will get an F
        System.out.println("\n");
        System.out.println(Test.pizzaParty()); //will print out if they get
a pizza party or dont
        System.out.println("\n");
```

```
        s1Test.retakeTest(s1); //allows s1 to retake the test if they get
below an 80
        System.out.println(Test.calculateAverage()); //prints the new
average after s1 retake
        System.out.println(s1); //prints out s1 final results
    }

}
```

Student.java:

```
import java.util.Scanner;
public class Student{
    //instance variables
    private static int numStudents = 0;
    private String name;
    private String date;
    private boolean reviewSubmitted;
    private Test t;

    private int score = 0;
    private static int totalScore;
    //default constructors
    public Student()
    {
        name = "";
        date = "";
        reviewSubmitted = false;
        numStudents++;
        t = new Test(5);
    }
    //overloaded constructors
    public Student(String name)
    {
        this.name = name;
        date = "";
        reviewSubmitted = false;
        numStudents++;
        t = new Test(5);
    }
```

```
    public Student(String name, String date)
    {
        this.name = name;
        this.date = date;
        reviewSubmitted = false;
        t = new Test();
        numStudents++;
    }
    public Student(String name, String date, boolean reviewSubmitted, Test
t)
    {
        this.name = name;
        this.date = date;
        this.reviewSubmitted = reviewSubmitted;
        numStudents++;
        this.t = t;
    }

    //mutator methods
    public void setName(String name)
    {
        this.name = name;
        numStudents++;
    }
    public void setDate(String date)
    {
        this.date = date;
    }
    public void setReview(boolean reviewSubmitted)
    {
        this.reviewSubmitted = reviewSubmitted;
    }
    public void setScore(int score)
    {
        this.score = score;
    }
    public void setTest(Test t)
    {
        this.t = t;
    }
```

```
public static void setNumOfStudents(int total)
{
    numStudents = total;
}
public static void setTotalScore(int ts)
{
    totalScore = ts;
}
//accessor methods
public String getName(){
    return name;
}
public String getDate()
{
    return date;
}
public boolean getReview(){
    return reviewSubmitted;
}
public Test getTest()
{
    return t;
}
public int getScore()
{
    return score;
}
public static int getNumStudents()
{
    return numStudents;
}
public static int getTotalScore()
{
    return totalScore;
}
//other methods
public void startTest(Test t) //this method contains the actual test
{
    Scanner scan = new Scanner(System.in);
    boolean answer;
```

```
        boolean q1 = false; //booleans are so the test questions do not get
repeated with the random number
        boolean q2 = false;
        boolean q3 = false;
        boolean q4 = false;
        boolean q5 = false;
        int count = 1;
        System.out.println(getName() + "\'s Test begins now: ");
        while(count <= Test.getQuestions())
        {
            int numOfQuestion = (int)(Math.random()*6); //chooses a random
question
            if(numOfQuestion == 1 && q1 == false)
            {
                System.out.println("True or False: OOP stand for Open Oven
Pizza? ");
                answer = false;
                System.out.println("Answer: ");
                boolean studentAnswer = scan.nextBoolean();
                if(answer == studentAnswer)
                {
                score++;
                }
                q1 = true;
                count++;
            }
            else if(numOfQuestion == 2 && q2 == false)
            {
                System.out.println("True or False: String a primitive data
type? ");
                answer = false;
                System.out.println("Answer: ");
                boolean studentAnswer = scan.nextBoolean();
                if(answer == studentAnswer)
                {
                score++;
                }
                q2 = true;
                count++;
            }
```

```
            else if(numOfQuestion == 3 && q3 == false)
            {
                System.out.println("True or False: == compares the content
in strings");
                answer = false;
                System.out.println("Answer: ");
                boolean studentAnswer = scan.nextBoolean();
                if(answer == studentAnswer)
                {
                score++;
                }
                q3 = true;
                count++;
            }
            else if(numOfQuestion == 4 && q4 == false)
            {
                System.out.println("True or False: Double is a primitive
data type");
                answer = true;
                System.out.println("Answer: ");
                boolean studentAnswer = scan.nextBoolean();
                if(answer == studentAnswer)
                {
                score++;
                }
                q4 = true;
                count++;
            }
            else if(numOfQuestion == 5 && q5 == false)
            {
                System.out.println("True or False: Private variables can
only be accessed in the same class");
                answer = true;
                System.out.println("Answer: ");
                boolean studentAnswer = scan.nextBoolean();
                if(answer == studentAnswer)
                {
                score++;
                }
                q5 = true;
```

```
                count++;
            }
            else {
                continue;
            }
            System.out.println("\n");

        }
        score = score * 100/Test.getQuestions(); //calculates the score as
a percentage
        totalScore += score; //adds this score to the total score (for the
average)

    }

    public void extraCredit(Student s){ //gives extra credit to the
students who have submitted the review homework
        if(s.getReview() == true)
            score+=5;
    }
    public Student highestScore(Student a, Student b) //returns student
with highest score
    {
        if(b.getScore() == a.getScore())
        {
            return null; //if both students have the same score
        }
        else if(b.getScore() >= a.getScore())
        {
            return b;
        }
       else if(this.getScore() >= a.getScore() && this.getScore() >=
b.getScore())
        {
            return this; //if there is another student
        }

        return a;

    }
```

```
    public String toString() //toString method
    {
        return ("Student's name: " + name + ", Date: " + date + ", Score: "
+ score);
    }
   }
```

Test.java

```
public class Test{
    private static final int a = 90; //variables defined for grading
purposes
    private static final int b = 80;
    private static final int c = 70;
    private static final int d = 60;
    private static final int f = 59;


    private String testVersion; //a, b, c
    private static int questions; //up till 5, static so that all students
have the same amount of questions (its fair)
    //default constructor
    public Test()
    {
        questions = 5;
        testVersion = "a";
    }
    //other constructors
    public Test(int numOfQuestions)
    {
        questions = numOfQuestions;
    }
    public Test(int numOfQuestions, String testVersion)
    {
        questions = numOfQuestions;
        this.testVersion = testVersion;
    }
    //mutator methods
    public static void setQuestions(int numOfQuestions)
    {
        questions = numOfQuestions;
```

```
    }
    public void setVersion(String testVersion)
    {
        this.testVersion = testVersion;
    }

    //accessor methods
    public static int getQuestions()
    {
        return questions;
    }
    public String getTestVersion()
    {
        return testVersion;
    }

    //other methods
    public static double calculateAverage() //calculates the average
between all the students
    {
        return (Student.getTotalScore() / Student.getNumStudents());
    }


    //overloaded methods
    public String calculateGrade() //if student does not input who they are
    {
        String letterGrade = "F";
        return "Did not provide name, your grade is: " + letterGrade;
    }
    public String calculateGrade(Student s) //calculates the letter grade
for the student
    {
        String letterGrade = "";
        if(s.getScore() >= a)
        {
            letterGrade = "A";
        }
        else if(s.getScore() >= b && s.getScore() <= a)
        {
```

```
            letterGrade = "B";
        }
        else if(s.getScore() >= c && s.getScore() <= b)
        {
            letterGrade = "C";
        }
        else if(s.getScore() >= d && s.getScore() <= c)
        {
            letterGrade = "D";
        }
        else if(s.getScore() <= f)
        {
            letterGrade = "F";
        }
        return s.getName() + " grade is: " + letterGrade;
    }

    //determines the test version
    public void determineTestVersion(Test t, Student s)
    {
        if(t.getTestVersion().equalsIgnoreCase("a"))
        {
            s.startTest(this);
        }
        else if(t.getTestVersion().equalsIgnoreCase("b"))
        {
            s.startTest(this);
        }
        else if(t.getTestVersion().equalsIgnoreCase("c"))
        {
            s.startTest(this);
        }
        else {
            System.out.println("Invalid test version");
            t.calculateGrade();
        }
    }
    //static method
    public static String pizzaParty() //if the average is above an 80,
students get a pizza party
```

```
    {
        String pizza = "";
        if(Test.calculateAverage() >= 80.0)
        {
            pizza = "Pizza party! :)";
        }
        else {
            pizza = "No pizza party :(";
        }
        return pizza;
    }

    public void retakeTest(Student s) //if student score is below 80%, they
can be allowed a retake
    {
        if(s.getScore() <= 80)
        {
            System.out.println("You can have a retake: ");
            int temp = Student.getTotalScore() - s.getScore();
            s.setTotalScore(temp);
            s.setScore(0);
            s.startTest(this);
        }
        else if(s.getScore() >= 80)
        {
            System.out.println("You are not allowed to have a retake");
        }
    }
    //toString
    public String toString()
    {
        return ("Class average: " + calculateAverage() + ", " +
Test.pizzaParty());
    }

}
```