```
public class Student
{
    //data members
    public final static int NUM_TESTS = 3;
    private String myName;
    private int[] myTests;
    private String myGrade;

    //constructors
    public Student()
    {
        myName = "";
        myTests = new int[NUM_TESTS];
        myGrade = "";
    }

    public Student(String name, int[] tests, String grade)
    {
        myName = name;
        myTests = tests;
        myGrade = grade;
    }

    public String getName()
    { return myName; }

    public String getGrade()
    { return myGrade; }

    public void setGrade(String newGrade)
    { myGrade = newGrade; }

    public void computeGrade()
    {
        if (myName.equals(""))
            myGrade = "No grade";
        else if (getTestAverage() >= 65)
            myGrade = "Pass";
        else
            myGrade = "Fail";
    }

    public double getTestAverage()
    {
        double total = 0;
        for (int score : myTests)
            total += score;
        return total/NUM_TESTS;
    }
}
```

```
public class UnderGrad extends Student
{
    public UnderGrad()   //default constructor
    { super(); }

    //constructor
    public UnderGrad(String name, int[] tests, String grade)
    { super(name, tests, grade); }

    public void computeGrade()
    {
        if (getTestAverage() >= 70)
            setGrade("Pass");
        else
            setGrade("Fail");
    }
}

public class GradStudent extends Student
{
    private int myGradID;

    public GradStudent()      //default constructor
    {
        super();
        myGradID = 0;
    }

    //constructor
    public GradStudent(String name, int[] tests, String grade,
            int gradID)
    {
        super(name, tests, grade);
        myGradID = gradID;
    }

    public int getID()
    { return myGradID; }
```

```
    public void computeGrade()
    {
        //invokes computeGrade in Student superclass
        super.computeGrade();
        if (getTestAverage() >= 90)
            setGrade("Pass with distinction");
    }
}
```