CT874

Assignment 6

H. Dip. Industry Stream

Hume, Tori (11486248)

Question 1

Code:

SUPERCLASS:

```
/*Tori Hume
 * ID: 11486248
 * Assignment 6
 * Question 1
 * Student SuperClass
 */


//create abstract class Student
abstract public class Student
{

	//declare variables and initialize test[]
	private String name, grade;
	private long ID;
	private int test[]= new int[NUM_TESTS];

	//declare and set constant
	private static final int NUM_TESTS=3;

	//create Student constructor
	public Student()
	{
		this.name= "Unavailable";
		this.ID= 0;
	}

	//create Student constructor with parameters
	public Student(String name, long ID)
	{
		this.name = name;
		this.ID = ID;
	}

	//getters and setters for each variable, and a getter for the NUM_TESTS constant

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

	public void setID(long ID) {
		this.ID = ID;
	}

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

	public void setTestScore(int i, int b){
		this.test[i]=b;
	}

	public int getTestScore(int i){
		return this.test[i];
	}

	public static int getNumTests() {
		return NUM_TESTS;
```

```
	}

	public String getName() {
		return name;
	}

	public long getID() {
		return ID;
	}

	public String getGrade() {
		return grade;
	}

	//override toString method
	@Override
	public String toString() {
		return "Student [Name= " + name + ", ID= " + ID + ", Grade=" + grade +  "]";
	}

	//abstract method calculateResuls
	abstract void calculateResult();


} //end of class
```

SUBCLASS':

```
/*Tori Hume
 * ID: 11486248
 * Assignment 6
 * Question 1
 * UnderGrad Subclass of Student
 */

//create subclass of Student.
public class UnderGrad extends Student{

	//create constructor
	public UnderGrad() {

		//pulls the constructor from the superclass Student
		super();

	}

	//Create constructor with parameters
	public UnderGrad(String name, long ID) {

		//pulls the constructor with parameters from the superclass Student
		super(name, ID);

	}

	//overriding and implementing the calculateResults method found
	//in the Superclass Student
	public void calculateResult(){

		//declaring and initialize
		int marks=0;

		//for loop used to iterate through the TestScore array
		for(int i=0; i<getNumTests(); i++){
			marks= marks + getTestScore(i);
```

```
		}

		//calculate the average Mark
		int averageMark=(marks/3);

		//decision used to set grade to pass or fail
		if( averageMark >= 40 ){
			setGrade(" PASS");
		}
		else{
			setGrade(" FAIL");
		}
	}
}


/*Tori Hume
 * ID: 11486248
 * Assignment 6
 * Question 1
 * PostGrad Subclass of Student
 */

//create subclass of Student.
public class PostGrad extends Student{

	//create constructor
	public PostGrad() {

		//pulls the constructor from the superclass Student
		super();
	}

	//Create constructor with parameters
	public PostGrad(String name, long ID) {

		//pulls the constructor with parameters from the superclass Student
		super(name, ID);
	}

	//overriding and implementing the calculateResults method found
	//in the Superclass Student

	public void calculateResult(){

		//declaring and initialize
		int marks=0;

		//for loop used to iterate through the TestScore array
		for(int i=0; i<getNumTests(); i++){
			marks= marks + getTestScore(i);
		}

		//calculate the average Mark
		int averageMark=(marks/3);

		//decision used to set grade to pass or fail
		if( averageMark >= 50 ){
			setGrade(" PASS");
		}
		else{
			setGrade(" FAIL");
		}
	}
}
```

CLIENT CLASS:

```
/*Tori Hume
 * ID: 11486248
 * Assignment 6
 * Question 1
 * Client class
 */

import java.util.Scanner;

public class Client
{
	public static void main(String[] args)
	{
		// create new Array of student classes called studentList
		Scanner input= new Scanner(System.in);
		Student[] studentList= new Student[3];
		int studentType;

		// create a for loop to populate classes and in turn the array
		for (int i = 0; i < studentList.length; i++) {

			//get user to select what type the student will be.
			System.out.println("Please select the type of student. \n\tEnter 1 for a "
					+ "PostGrad student. \n\tEnter 2 for a Undergrad Student");

			studentType=input.nextInt();

			input.nextLine();


			// using an if and else statement to implement the decision of postgrad or
			undergrad
			if (studentType == 1){

				//create a new instance of postgrad called "a"
				PostGrad a = new PostGrad();

				//Use input & setters to assign values entered in the comand window
				to variables in "a"
				System.out.println("Enter Student Name");
				String n=input.nextLine();
				a.setName(n);

				System.out.println("Enter Student ID");
				long id=input.nextLong();
				a.setID(id);

				//use a for loop & setters to fill the testScore array
				for(int j=0; j<Student.getNumTests(); j++){

					System.out.println("Enter Results from test "+ (j+1));
					int r=input.nextInt();
					a.setTestScore(j,r);
				}

				//call calculateResuts method for "a"
				a.calculateResult();
```

```
                        //Print toString method to screen
                        System.out.println(a.toString());
                        System.out.println("\n");
                        //assign the instance "a" of PostGrad to the ith index of
                        studentList
                        studentList[i]=a;
                }

                else {
                        //create a new instance of UnderGrad called "a"
                        UnderGrad a = new UnderGrad();

                        //Use input & setters to assign values entered in the command window
                        to variables in "a"
                        System.out.println("Enter Student Name");
                        String n=input.nextLine();
                        a.setName(n);

                        System.out.println("Enter Student ID");
                        long id=input.nextLong();
                        a.setID(id);

                        //use a for loop & setters to fill the testScore array
                        for(int j=0; j<Student.getNumTests(); j++){

                                System.out.println("Enter Results from test "+ (j+1));
                                int r=input.nextInt();
                                a.setTestScore(j,r);
                        }
                        //call calculateResuts method for "a"
                        a.calculateResult();
                        //Print toString method to screen
                        System.out.println(a.toString());
                        System.out.println("\n");
                        //assign the instance "a" of PostGrad to the ith index of
                        studentList
                        studentList[i]=a;
                }
        }
        input.close();

    }

}
```

Screen Shot:

```
<terminated> Client [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (26 Oct 2016, 17:18:48)
Please select the type of student.
        Enter 1 for a PostGrad student.
        Enter 2 for a Undergrad Student
1
Enter Student Name
Tara Kelly
Enter Student ID
101
Enter Results from test 1
40
Enter Results from test 2
40
Enter Results from test 3
45
Student [Name= Tara Kelly, ID= 101, Grade= FAIL]


Please select the type of student.
        Enter 1 for a PostGrad student.
        Enter 2 for a Undergrad Student
1
Enter Student Name
Rick Jones
Enter Student ID
102
Enter Results from test 1
70
Enter Results from test 2
65
Enter Results from test 3
65
Student [Name= Rick Jones, ID= 102, Grade= PASS]


Please select the type of student.
        Enter 1 for a PostGrad student.
        Enter 2 for a Undergrad Student
2
Enter Student Name
Steven Larken
Enter Student ID
103
Enter Results from test 1
70
Enter Results from test 2
50
Enter Results from test 3
65
Student [Name= Steven Larken, ID= 103, Grade= PASS]
```