**Supplemental Lab Sheet**

Read the following description and provide the 3 missing Java files.

A school uniform supplier for primary and secondary schools has 2 different types of uniforms, regular school uniforms (for class) and sports uniforms (for PE). Each school has a different crest (a name) and colour for their regular uniforms. They also may specify an optional coat for their uniforms. Only secondary schools have sports uniforms, and all sports uniforms have a material quality rating and also may specify an optional set of sports footwear (such as Gaelic football boots or running track shoes). The supplier has hired you to write a program that allows the input of the uniform details for each school, calculate the unit cost per uniform and total cost of all uniforms for the school.

There are 3 classes required: an abstract class **uniform** which has member variables for school name and whether it is a primary school uniform. **Uniform** has an abstract method called *calculateCost()*.

In Java, add the following details to Uniform:

* Make it abstract
* Give it an abstract method calculateCost()
* Give it 2 private fields, name and isPrimary
* Give it getters for the fields
* Give it a constructor which initialises the fields.

Uniform has 2 concrete subclasses, **regular uniform** and **sports uniform**.

Add the following details to **RegularUniform**:

* Give it 3 private fields, colour (string), howMany(int) and coatRequired(boolean)
* Give it a constructor which passes values to the parent and sets its own values
* A getter for howMany
* Implement calculateCost as follows: first print a message "Calculating cost for " + colour + " uniform with " + getName() + " on the crest". The cost is 50 euros + the number of letters in the crest (use String API) multiplied by 1.5. If a coat is required, it's an extra 25 euro. Return the cost to the test program which called it

Add the following details to **SportsUniform**:

* Add private fields materialQuality (int) and footwearRequired (boolean)
* Add a constructor that takes in and passes on the parent class fields and sets its own fields
* implement inherited abstract method calculateCost() as follows: cost is 25 + (materialQuality \* 0.5). If footwearRequired is true, 15 euro is added to the cost. Return the cost to the calling test class.

The test class is available on Moodle. The sample output is below. Try to match it exactly.

What is the name of the school?

Hogwarts

Is this a primary school?

false

What colour should the school uniform be?

black

How many of these uniforms will you need?

100

Does your uniform require an extra coat? true/false

true

Calculating cost for black uniform with Hogwarts on the crest

The unit cost of this school uniform is: 87.0 each.

The total cost of all the school uniforms is: 8700.0

Do you require a sports uniform?

true

What quality of material do you want (0 - 100):

50

Does your uniform require an special footwear? true/false

true

The unit cost of this sports uniform order is: 65.0

The total cost of all the sports uniforms is: 6500.0

Do you wish to process another school uniform order? true/false

true

What is the name of the school?

St. Trinians

Is this a primary school?

true

What colour should the school uniform be?

red

How many of these uniforms will you need?

10

Does your uniform require an extra coat? true/false

false

Calculating cost for red uniform with St. Trinians on the crest

The unit cost of this school uniform is: 68.0 each.

The total cost of all the school uniforms is: 680.0

Do you wish to process another school uniform order? true/false

false

Total cost for all schools was 15880.0