Practical 8

Task 1: (20 marks)

For this Task you are required to design a piza ordering system for 2 major Pizza store Debonairs and Romans.

You are provided with the details of the base Pizza class:

Attributes	
baseType	type of base of the pizza
size	The size of the pizza
basePrice	The price of the pizza with no toppings
Methods	
makeBase()	This method will specify the type of pizza base
addToppings()	Adds toppings to the pizza
setSize()	Sets the size of the pizza
calculatePrice()	Calculates the price based on the size and toppings

In this scenario, Debonairs only makes stuffed crust pizza bases and Romans only make Pan pizza bases override the necessary function to achieve the necessary changes.

Additional:

- All toppings will be an additional R15

- Pizza price per size:
    - Small: R50
    - Medium: R75
    - Large: R100

You are required to implement the required header files.
You will be provided with the following files where you need to define your objects:

Pizza.cpp	Pizza.h
RomansPizza.cpp	RomansPizza.h
DebonairsPizza.cpp	DebonairsPizza.h

A main is provided to you to test if your implementation is correct
Below are a few expected input and outputs you should display:

Example 1:

Input:

```
Select Pizza Store:
1. Debonairs (Stuffed Crust)
2. Romans (Pan Pizza)
Choice: 1

Select Pizza Size (Small/Medium/Large): Medium

How many toppings would you like? 3
```

Output:

```
Pizza Base: Stuffed Crust
Size: Medium
Number of Toppings: 3
Total Price: R120
```

Example 2:

Input:

```
Select Pizza Store:
1. Debonairs (Stuffed Crust)
2. Romans (Pan Pizza)
Choice: 2

Select Pizza Size (Small/Medium/Large): Large

How many toppings would you like? 1
```

Output:

```
Pizza Base: Pan Pizza
Size: Large
Number of Toppings: 1
Total Price: R115
```

Example 3:

Input:

```
Select Pizza Store:
1. Debonairs (Stuffed Crust)
2. Romans (Pan Pizza)
Choice: 2

Select Pizza Size (Small/Medium/Large): Small

How many toppings would you like? 0
```

Output:

```
Pizza Base: Pan Pizza
Size: Small
Number of Toppings: 0
Total Price: R50
```

Task 2: (30 marks)

Observe the table below:

Class	Attributes	Description of Attribute		Methods	Description of Methods
Parent Class					
BMWCar	model	The model name of the car		showDetails()	Displays the car model and base price
	BasePrice	The base price of the car		getPrice()	Returns the total price of the car
Child Classes					
Sedan	hasSunroof	Flag indicating whether the car has a sunroof		addSunroof()	Sets the hasSunroof flag to true, indicating that the car has a sunroof
				getPrice()	Returns the base price of the car with an additional amount if the car has a sunroof
				showDetails()	Displays the car model, base price, and whether the car has a sunroof
SUV	hasAllWheelDrive	Flag indicating whether the car has all-wheel drive		addAllWheelDrive()	Sets the hasAllWheelDrive flag to true, indicating that the car has all-wheel drive
				getPrice()	Returns the base price of the car with an additional amount if the car has all-wheel drive
				showDetails()	Displays the car model, base price, and whether the car has all-wheel drive

HatchBack	hasSportPackage	Flag indicating whether the car has a sport package		addSportPackage()	Sets the hasSportPackage flag to true, indicating that the car has a sport package
				getPrice()	Returns the base price of the car with an additional amount if the car has a sport package
				showDetails()	Displays the car model, base price, and whether the car has a sport package
Coupe	hasLeatherSeats	Flag indicating whether the car has leather seats		addLeatherSeats()	Sets the hasLeatherSeats flag to true, indicating that the car has leather seats
				getPrice()	Returns the base price of the car with an additional amount if the car has leather seats
				showDetails()	Displays the car model, base price, and whether the car has leather seats

You are tasked to create the above classes for a BMW ordering car system.
Here are the details of the cars:

Type	Base Price	Additional feature	Additional feature price
Sedan	R500 000	Sunroof	R15 000
SUV	R750 000	All-Wheel drive	R20 000
Hatchback	R400 000	Sport package	R10 000
Coupe	R550 000	Leather seat	R12 000

The dealership offers different types of BMW cars: Sedan, SUV, Hatchback, and Coupe. Each car type has unique features. You will implement a system that allows users to select a car type, choose additional features, and display the total price.

BMWCar.h	BMWCar.cpp
Coupe.h	Coupe.cpp
Hatchback.h	Hatchback.cpp
SUV.h	SUV.cpp
Sedan.h	Sedan.cpp
Task2.cpp	

You are also required to write a main for the program
which will out the necessary details
Use the examples below as a guide:

Example 1:

Input:

```
Select a car model:
1. BMW Sedan
2. BMW SUV
3. BMW Hatchback
4. BMW Coupe
Choice: 1

Would you like to add a sunroof for R15,000? (1. Yes / 2. No): 1
```

Output:

```
Model: BMW Sedan
Base Price: R500000
Features: Sunroof (R15,000 extra)
Total Price: R515000
```

Example 2:

Input:

```
Select a car model:
1. BMW Sedan
2. BMW SUV
3. BMW Hatchback
4. BMW Coupe
Choice: 2

Would you like to add all-wheel drive for R20,000? (1. Yes / 2. No): 1
```

Output:

```
Model: BMW SUV
Base Price: R750000
Features: All-Wheel Drive (R20,000 extra)
Total Price: R770000
```

Example 3:

Input:

```
Select a car model:
1. BMW Sedan
2. BMW SUV
3. BMW Hatchback
4. BMW Coupe
Choice: 3

Would you like to add a sport package for R10,000? (1. Yes / 2. No): 2
```

Output:

```
Model: BMW Hatchback
Base Price: R400000
Total Price: R400000
```

Example 4:

Input:

```
Select a car model:
1. BMW Sedan
2. BMW SUV
3. BMW Hatchback
4. BMW Coupe
Choice: 4

Would you like to add leather seats for R12,000? (1. Yes / 2. No): 1
```

Output:

```
Model: BMW Coupe
Base Price: R550000
Features: Leather Seats (R12,000 extra)
Total Price: R562000
```

Example 5:

Input:

```
Select a car model:
1. BMW Sedan
2. BMW SUV
3. BMW Hatchback
4. BMW Coupe
Choice: 2

Would you like to add all-wheel drive for R20,000? (1. Yes / 2. No): 2
```

Output:

```
Model: BMW SUV
Base Price: R750000
Total Price: R750000
```