Coffee Shop Challenge

You are a member of a team to design and implement a simple point-of-sale (POS) solution for a client who owns a café. The café is best known for its wide range of coffees.

Your job, as developer, is to design the domain model. Specifically:

- An object-oriented implementation of the classes and interfaces which best model the different types of coffees, sizes, and condiments
- The business logic to calculate the final cost of an order.

Note: The UI and other elements of the implementation is being handled by other team members.

The POS solution takes in the 3 or more selections and calculate the final cost (for example 1 Large House blend Espresso with milk, 1 Addict Dark roast Latte with sugar, 1 Robusta Cappuccino). Your job, as developer, is to write the server-side code to calculate the final cost. You need to design and write an object-oriented solution. You will need to figure out how to represent the different selections and then develop an interface to perform the calculations.

The following is a list of types of coffees and their base prices:

- *House blend ($1)*
- *Dark roast ($1.50)*
- *Robusta ($2.00)*
- *Arabica ($2.50)*

The coffee can be prepared in the following ways:

- *Espresso (+ $1.00)*
- *Latte (+ $1.25)*
- *Cappuccino (+ $1.50)*
- *Macchiato (+ $1.75)*
- *Mocha (+ $2.00)*

Coffees can then be served in the following sizes:

- *Standard (no change)*
- *Child (x 0.75)*
- *Large (x 1.5)*
- *Addict (x 2)*

In addition, the following additional optional condiments (1 or more) can be added:

- *Milk (+ $1)*
- *Sugar (+ $0.25)*
- *Coco powder (+ $0.10)*

Your task is to develop the point-of-sale application which enters the following selections:

- Coffee type
- Preparation style
- Size
- 1 or more optional condiment

Evaluation Criteria:

- Your design should be flexible to add new coffee types, preparation methods, sizes, and additional condiments.
- Quality control - how do you know the final cost is correct and what checks have you put in place?
- You should provide reasons for your choice of design.

You may write the solution in an object-oriented language of your choice (e.g., Java, C#, C++). An initial starter solution can be created in Java as a Maven project (refer to https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html).