

OOP Master Challenge Exercise

The purpose of the application is to help a fictitious company called Bills Burgers to manage their process of selling hamburgers.

Our application will help Bill to select types of burgers, some of the additional items (additions) to be added to the burgers and pricing.

We want to create a base hamburger, but also two other types of hamburgers that are popular ones in Bills store.

The basic hamburger should have the following items:

- Bread roll type, meat and up to 4 additional additions (things like lettuce, tomato, carrot, etc) that the customer can select to be added to the burger,
- Each one of these items gets charged an additional price so you need some way to track how many items got added and to calculate the final price (base burger with all the additions).
- This burger has a base price and the additions are all separately priced (up to 4 additions, see above).

Create a Hamburger class to deal with all the above. The constructor should only include the roll type, meat and price, can also include name of burger or you can use a setter.

Also create two extra varieties of Hamburgers (subclasses) to cater for:

- a) Healthy burger (on a brown rye bread roll), plus two addition items that can be added. The healthy burger can have 6 items (Additions) in total. **Hint:** *you probably want to process the two additional items in this new class (subclass of Hamburger), not the base class (Hamburger), since the two additions are only appropriate for this new class (in other words new burger type),*
- b) Deluxe hamburger - comes with chips and drinks as additions, but no extra additions are allowed. **Hint:** *You have to find a way to automatically add these new additions at the time the deluxe burger object is created, and then prevent other additions being made.*

All 3 classes should have a method that can be called anytime to show the base price of the hamburger plus all additionals, each showing the addition name, and addition price, and a grand/final total for the burger (base price + all additions).

For the two additional classes this may require you to be looking at the base class for pricing and then adding totals to final price.