

OOP Project Overview

The **OOP Project** is an opportunity for bootcamp students to:

- **Apply key concepts:** Pull together various topics learned so far, such as classes and polymorphism, user input, arrays and lists, and exception handling.
- **Work on a larger case study:** This project is more complex than lab exercises, serving as a precursor to the Final Project.
- **Collaborate in a team:** Experience working on a software project that requires proper object-oriented design, documentation, and version control.

Every student will participate in this project as part of an assigned group. Most of the class time will be dedicated to working on the project, though some outside work may be necessary. Regular check-ins will ensure groups stay on track.

Each group will choose one of the following three projects. Once your group decides on a project, stick with it—no switching after starting!

Possible Projects

1. **Point-of-Sale Terminal**
A cash register/ordering terminal for a retail setting like a store, coffee shop, or fast-food restaurant.
2. **Library System**
A system to manage borrowing, returning, and tracking library materials.
3. **Fitness Center**
A management system for members, classes, and facilities in a fitness center.

Each project description outlines the minimum requirements. It's encouraged for groups to go beyond these and add features of interest. Let your instructors know which project you choose and share your GitHub repo link as soon as you have it.

Point-of-Sale Terminal

Build a cash register or self-service terminal for a retail environment, such as a small store, coffee shop, or fast-food restaurant.

Requirements:

- Create a **Product** class that includes:
 - Name
 - Category

- Description
 - Price
 - Store at least 12 items in a list.
- Display a **menu** for the user to select items (by number or letter).
 - Allow the user to choose a **quantity** for each item ordered.
 - Calculate and display the **line total** (item price \times quantity).
- Offer options to:
 - Re-display the menu.
 - **Complete the purchase.**
- Calculate and display:
 - **Subtotal**
 - **Sales tax**
 - **Grand total** (Use the Math library to handle rounding issues).
- Ask for the **payment type**:
 - **Cash**: Ask for the amount tendered and provide change.
 - **Check**: Ask for the check number.
 - **Credit**: Ask for the credit card number, expiration, and CVV.
- At the end, display a **receipt** with:
 - All items ordered.
 - Subtotal, grand total.
 - Payment details.
- Return to the original menu for a new order.

Hint: Use an array or list to keep track of ordered items.

Optional Enhancements:

- **Moderate:** Store your list of products in a text file. Add an option to append new items to the product list and update the file.
- **Buff:** Do a push-up each time you encounter an exception or error while running your code!

Build a cash register or self-service terminal for a retail environment, such as a small store, coffee shop, or fast-food restaurant.

Main files

Classes

Methods

Program.cs

Product.cs
- Number
- Name
- Category
- Description
- Price
- Item Dictionary (atleast 12 items)
- Maybe a Separate Enum – Maybe a text file if there is time

Use Switch case similar to kendall exercise folder – to choose items to add to the list from dictionary list

Display Item List

Create Purchase List

Display Purchase List

PaymentType.cs

-Cash
-Check
-Credit

Maybe a switch cases

Cash

Ask user for amount they will pay and then provide them with change + receipt

Check

Ask user for check number – maybe store it in a list or txt log? + receipt

Card

Ask user for CC#, Expiry, CVV – maybe store it in a txt log + receipt

Receipt.cs

- Sub Total
- Sales Tax
- Grand Total
- *Change
- Payment Details
- Item list

Make this abstract so we can use poly

Display Receipt

Calculate:

- SubTotal
- Sales Tax
- Grand Total