```cpp
#include <iostream>
#include <conio.h>
#include <cstdlib>

using namespace std;

/************** Cash Module **************
    This module is responsible for handel-
    ing the cash counter or cash register.
*****************************************/
class CashModule{
    public:
        CashModule(): cash(500) { }
        CashModule(int c): cash(c) {}
        int currentBalance();
        void acceptAmount(int);
    private:
        int cash;

};
/************ Dispenser Type *************
    Dispenser  type  class will define the
    items in the store. How much they cost
    and many items we have of that type.
*****************************************/
class DispenserType{
    public:
        // Constructors
        DispenserType(): cost(100), itemCount(100) { }
        DispenserType(int c, int i): cost(c), itemCount(i) { }

        // Member functions
        int getItems();
        int getCost();
        void makeSale(int);
    private:
        // Data Members
        int itemCount;
        int cost;
};

// Function Prototype
void showSelectionItems();
void sellProduct(CashModule &cashRegister, DispenserType &product);
void pressEntertoClear();

int main(){
    /*************************************
        Choice  variable  will  store  the
        selection of customer from the menu.
    *************************************/
    int choice;
```

```
    // These are the Items in our store.
    DispenserType  sweets(150,1), chips(50,120),
                   gum(5,300),cookies(10, 200),bisconni(30,70);
    /*************************************
        It  create  a  cash  register  with
        default  value  of  100  which will
        store the cash from customers.
    *************************************/
    CashModule cashRegister;

    /*****************************************
        This do while  loop  will  keep asking
        to purchase from  our  customer untill
        he/she presses 0 to exit from our shop.
    *****************************************/
    do{
        showSelectionItems();
        cout << "Choose: "; cin >> choice;
        switch(choice){
            case 1:
                // Sweets
                sellProduct(cashRegister, sweets);
                break;
            case 2:
                // Chips
                sellProduct(cashRegister, chips);
                break;
            case 3:
                // Gum
                sellProduct(cashRegister, gum);
                break;
            case 4:
                // Bisconni
                sellProduct(cashRegister, bisconni);
                break;
            case 5:
                // Cookies
                sellProduct(cashRegister, cookies);
                break;
        }
    }while(choice!=0);

    return 0;
}
/*********** Cash Module ***********
    Member function's definitions
***********************************/
int CashModule::currentBalance(){
    return cash;
}
void CashModule::acceptAmount(int c){
    cash+=c;
}
```

```
/********** Dispenser Type **********
    Member function's definitions
************************************/
int DispenserType::getItems(){
    return itemCount;
}
int DispenserType::getCost(){
    return cost;
}
void DispenserType::makeSale(int num=1){
    itemCount-=num;
}

/********* Helping Functions********
    These  functions are  helping
    functions  as  their  purpose
    of creation is to divide small
    repeating tasks into functions.
************************************/
void showSelectionItems(){
    // This function displays the items to the customer.
    cout << "1 - Sweets" << endl;
    cout << "2 - Chips" << endl;
    cout << "3 - Gum" << endl;
    cout << "4 - Bisconni" << endl;
    cout << "5 - Cookies" << endl;
    cout << "0 - Exit" << endl;
}

void sellProduct(CashModule &cashRegister, DispenserType &product){
    // This function sells the product and
    // show the results if successful or not.
    int cash;
    if(product.getItems()>0){
        cout << "Price: Rs."<< product.getCost() << endl;
        cout << "Cash: "; cin >> cash;
        if(cash>=product.getCost()){
            product.makeSale();
            cashRegister.acceptAmount(product.getCost());
            if(cash > product.getCost()){
                cout << "Cash Return: " << (cash - product.getCost()) << endl;
            }
            cout << "Item Purchased. Thanks for purchasing from Al-Jannat Store." << endl;
        }
        else{
            cout << "Insufficient Cash Amount." << endl;
        }
    }
    else{
        cout << "Item is out of stock." << endl;
    }
    pressEntertoClear();
}
```

```cpp
void pressEntertoClear(){
    /**************************************************
        This  function  is  made  to have pause
        when the customers  purchase items and
        wait for them to press enter to clear screen.
    **************************************************/
    cout << "Press enter to continue....";
    getch();
    system("cls");
}
```

OUTPUT

```
┌──(vex㉿Revolve)-[/mnt/c/Users/Vex/Github/university_work/3. OOP/Lab/000.Assignments/2]
└─$ g++ q1.cpp -o q1 && ./q1
1 - Sweets
2 - Chips
3 - Gum
4 - Bisconni
5 - Cookies
0 - Exit
Choose: 1
Price: Rs.150
Cash: 300
Cash Return: 150
Item Purchased. Thanks for purchasing from Al-Jannat Store.
```

Next time after pressing enter:

```
1 - Sweets
2 - Chips
3 - Gum
4 - Bisconni
5 - Cookies
0 - Exit
Choose: 1
Item is out of stock.
```