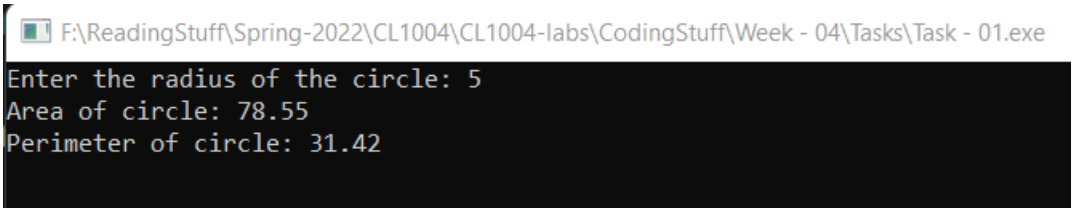
LAB TASKS:

Task - 01:

Your task is to create a Circle constructor that creates a circle with a radius provided by an argument. The circles constructed must have two getters getArea() (PIr^2) and getPerimeter() (2PI*r) which give both respective areas and perimeter (circumference).

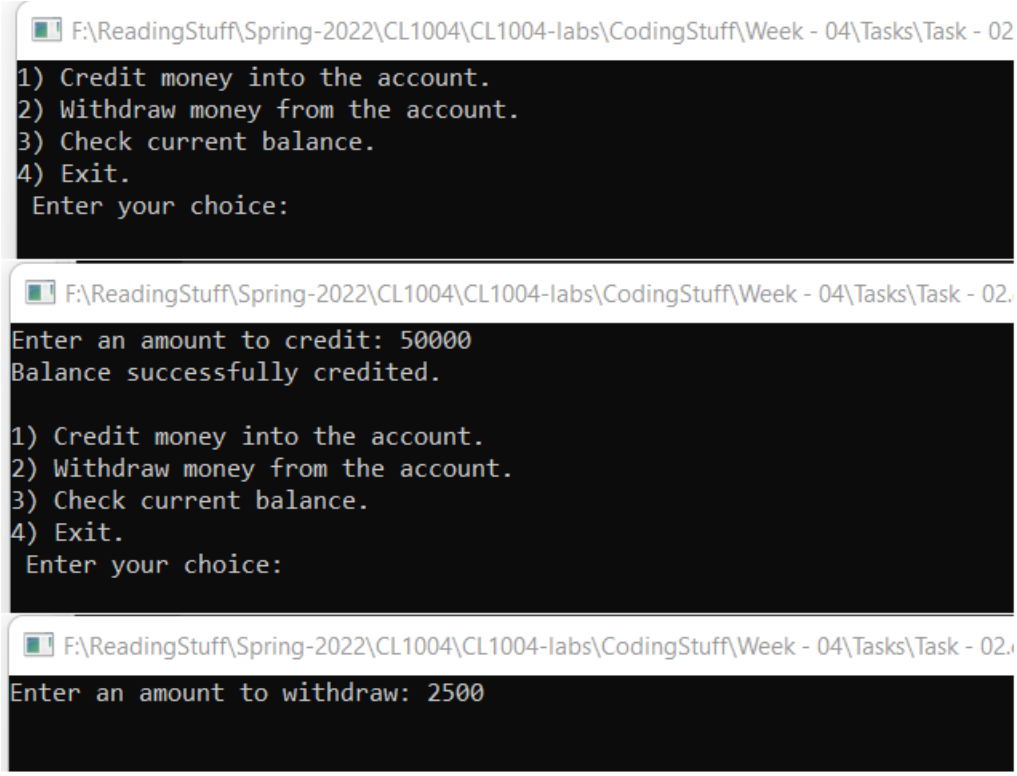
Sample output:



Task - 02:

Create an Account class that a bank might use to represent customers bank accounts. Include a data member to represent the account balance. Provide three member functions. Member function credit should add an amount to the current balance. Member function debit should withdraw money from the Account. Member function get Balance should return the current balance.
The screen must be cleared after each choice.

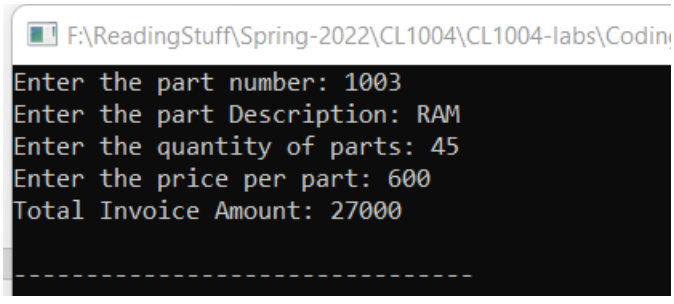
Sample output:



Task - 03:

Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four data members a part number, a part description, a quantity of the item being purchased and a price per item.

Sample output:



Task - 04:

A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise "Required copies not in stock" is displayed.

Design a system using a class called books with suitable member functions and constructors.

Task - 05:

Write a class called CoffeeShop, which has three instance variables:

Name: a string (basically, of the shop)

Menu: an array of items (of type MenuItem), with each item containing the item (name of the item), type (whether a food or a drink) and price.

Orders: an empty array and seven methods:

addOrder: adds the name of the item to the end of the orders array if it exists on the menu. Otherwise, return "This item is currently unavailable!"

fulfillOrder: if the orders array is not empty, return "The {item} is ready!". If the orders array is empty, return "All orders have been fulfilled!"

listOrders: returns the list of orders taken, otherwise, an empty array.

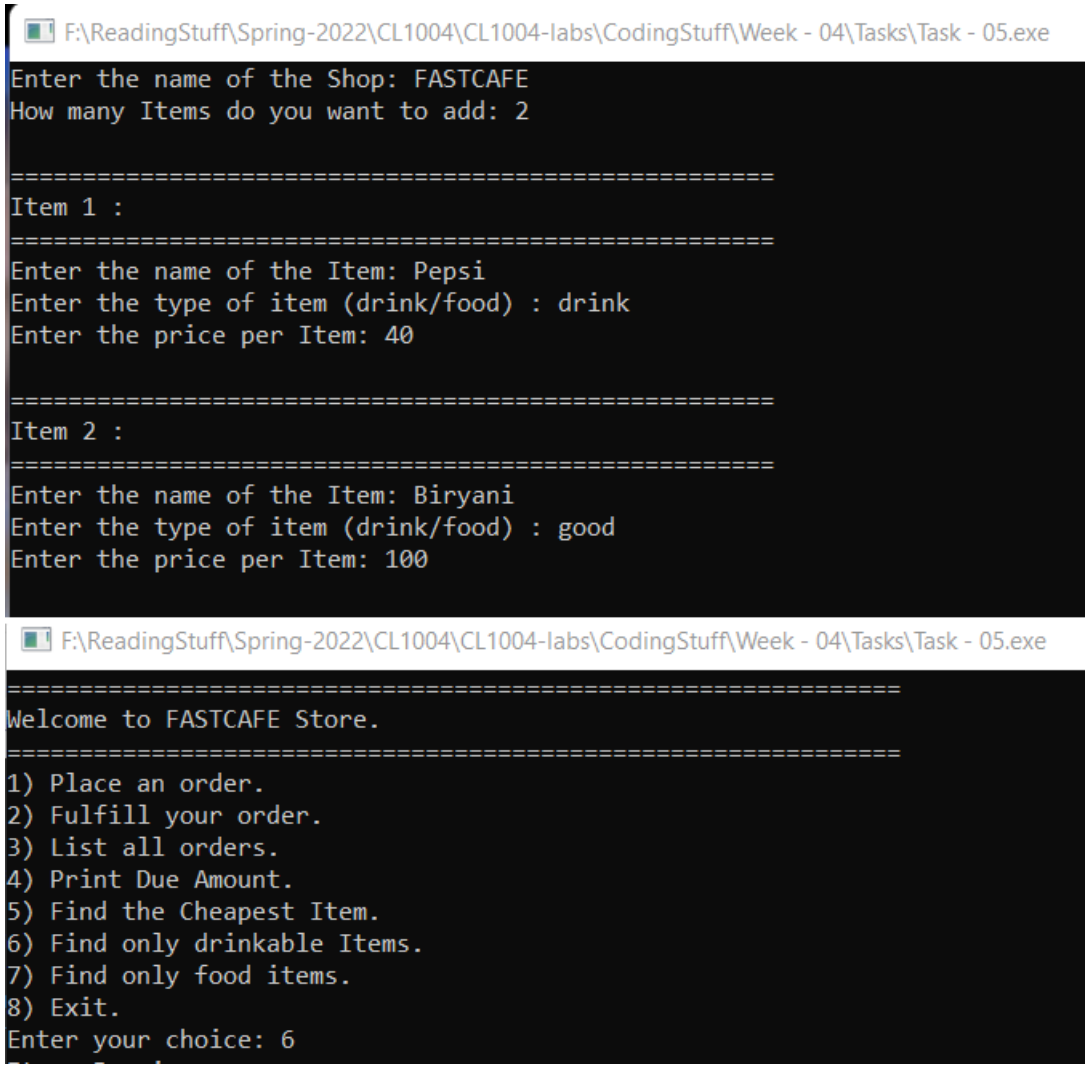
dueAmount: returns the total amount due for the orders taken.

cheapestItem: returns the name of the cheapest item on the menu.

drinksOnly: returns only the item names of type drink from the menu.

foodOnly: returns only the item names of type food from the menu.

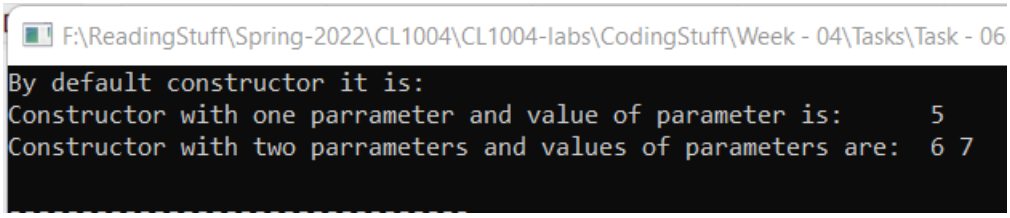
Sample output:



Task - 06:

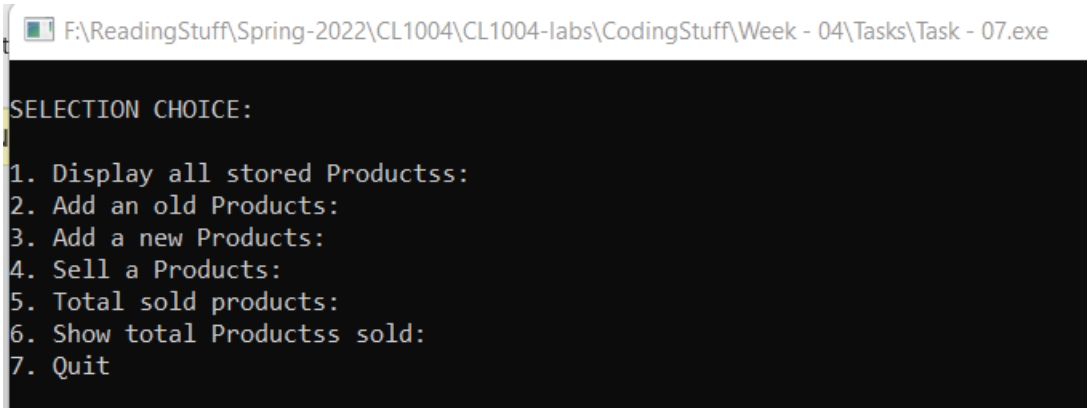
Write a program which consist of a class name PlayGround having two members length and width. The class must consist of three constructors which will be a constructor with two parameters, a constructor with parameter and one default constructor.

Sample output:



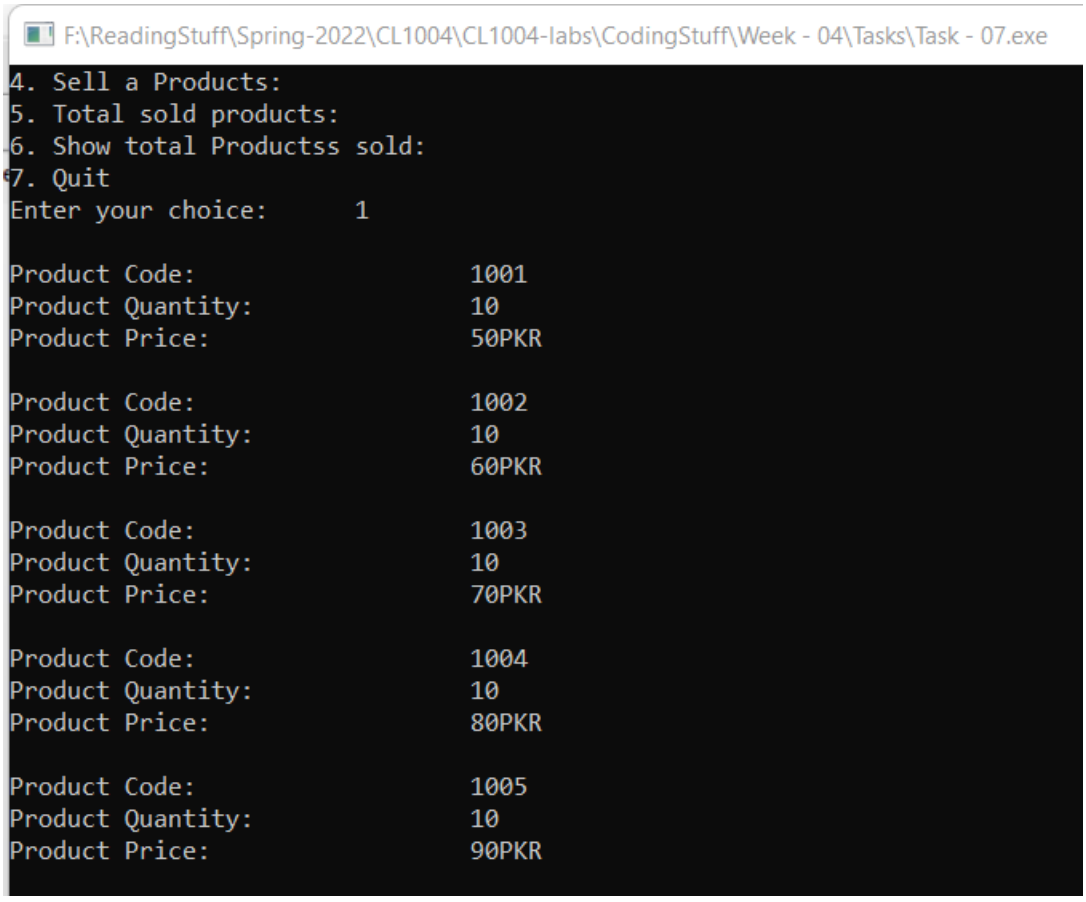
Task - 07:

You are assigned a task by your instructor to design an application for your café owner for keeping the record of five different products only. The café owner wants to store the products by a code each product along with quantity and their price. He is further interested in some more features for the mentioned application and he provides a following screenshot which contains the list of features.



Your task to write an object-oriented application for the mentioned application.

Sample output:



Task – 08:

Your instructor asked you to design an application for him where he can store information about different tv seasons like season name, number episodes and duration of particular episode. Your written object-oriented application output screen may be as follows:

Sample output: