

Assignment 4

1. Employee Class

Write a class named Employee that has the following member variables:

- **name.** A string that holds the employee's name
- **idNumber.** An int variable that holds the employee's ID number
- **department.** A string that holds the name of the department where the employee works.
- **position.** A string that holds the employee's job title.

The class should have the following constructors:

- A constructor that accepts the following values as arguments and assigns them to the appropriate member variables: employee's name, employee's ID number, department, and position.
- A constructor that accepts the following values as arguments and assigns them to the appropriate member variables: employee's name and ID number. The department and position fields should be assigned an empty string ("").
- A default constructor that assigns empty strings ("") to the name, department, and position member variables, and 0 to the idNumber member variable.

Write appropriate mutator functions that store values in these member variables and accessor functions that return the values in these member variables. Once you have written the class, write a separate program that creates three Employee objects to hold the following data.

Name	ID Number	Department	Position
Susan Meyers	47899	Accounting	Vice President
Mark Jones	39119	IT	Programmer
Joy Rogers	81774	Manufacturing	Engineer

The program should store this data in the three objects and then display the data for each employee on the screen.

2. Car Class

Write a class named Car that has the following member variables:

- **yearModel.** An int that holds the car's year model.
- **make.** A string that holds the make of the car
- **speed.** An int that holds the car's current speed.

In addition, the class should have the following constructor and other member functions.

- **Constructor.** The constructor should accept the car's year model and make as arguments. These values should be assigned to the object's yearModel and make member variables. The constructor should also assign 0 to the speed member variables.

- **Accessor.** Appropriate accessor functions to get the values stored in an object's yearModel, make, and speed member variables.
- **accelerate.** The accelerate function should add 5 to the speed member variable each time it is called.
- **brake.** The brake function should subtract 5 from the speed member variable each time it is called.

Demonstrate the class in a program that creates a Car object, and then calls the accelerate function five times. After each call to the accelerate function, get the current speed of the car and display it. Then, call the brake function five times. After each call to the brake function, get the current speed of the car and display it.

3. RetailItem Class

Write a class named RetailItem that holds data about an item in a retail store. The class should have the following member variables:

- **description.** A string that holds a brief description of the item
- **unitsOnHand.** An int that holds the number of units currently in inventory.
- **price.** A double that holds the item's retail price

Write a constructor that accepts arguments for each member variable, appropriate mutator functions that store values in these member variables, and accessor functions that return the values in these member variables. Once you have written the class, write a separate program that creates three RetailItem objects and stores the following data in them.

	Description	Units On Hand	Price
Item #1	Jacket	12	59.95
Item #2	Designer Jeans	40	34.95
Item #3	Shirt	20	24.95

4. Inventory Class

Design an Inventory class that can hold information and calculate data for items in a retail store's inventory. The class should have the following *private* member variables:

Variable Name	Description
itemNumber	An int that holds the item's item number
quantity	An int for holding the quantity of the items on hand
cost	A double for holding the wholesale per-unit cost of the item
totalCost	A double for holding the total inventory cost of the item (calculated as quantity times cost)

The class should have the following *public* member functions:

Member Function	Description
-----------------	-------------

Default Constructor	Sets all the member variables to 0
Constructor #2	Accepts an item's number, cost, and quantity as arguments. The function should copy these values to the appropriate member variables and then call the <code>setTotalCost</code> function.
<code>setItemNumber</code>	Accepts an integer argument that is copied to the <code>itemNumber</code> member variable.
<code>setItemNumber</code>	Accepts an integer argument that is copied to the <code>quantity</code> member variable.
<code>setCost</code>	Accepts a double argument that is copied to the <code>cost</code> member variable.
<code>setTotalCost</code>	Calculates the total inventory cost for the item (<code>quantity</code> times <code>cost</code>) and stores the result in <code>totalCost</code> .
<code>getItemNumber</code>	Returns the value in <code>itemNumber</code> .
<code>getQuantity</code>	Returns the value in <code>quantity</code> .
<code>getCost</code>	Returns the value in <code>cost</code> .
<code>getTotalCost</code>	Returns the value in <code>totalCost</code> .

Demonstrate the class in a driver program.

Input Validation: Do not accept negative values for item number, quantity, or cost.