Object-oriented Programming – C++

Tutorial 4

1. 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 a constructor that accepts the following values as arguments and assigns them to the appropriate member variables: employee's name, ID number, department, and position.

   Write appropriate mutator functions (setter) that store values in these member variables and accessor functions (getter) that return the values in these member variables. Once you have written the class, use a main 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. Design an Inventory class to 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**
Constructor	Accepts an item's number, cost, and quantity as arguments. The function should copy these values to the appropriate member variables and call the setTotalCost function.
setItemNumber	Accepts an integer argument that is copied to the itemNumber member variable.
setQuantity	Accepts an integer argument that is copied to the quantity member variable.
setCost	Accepts a double argument that is copied to the cost member variable.
setTotalCost	Calculates the total inventory cost for the item (quantity * cost) and stores the result in totalCost.
getItemNumber	Returns the value in itemNumber.
getQuantity	Returns the value in quantity.
getCost	Returns the value in cost .
getTotalCost	Returns the value in totalCost.

Demonstrate the class in a main program.

3. Create an Account class that a bank might use to represent customers' bank accounts. Include a data member of type int to represent the account balance. Provide a constructor that receives an initial balance and uses it to initialize the data member. The constructor should validate the initial balance to ensure it's greater than or equal to 0. If not, set the balance to 0 and display an error message indicating that the initial balance was invalid.

   Provide three member functions. Member function credit should add an amount to the current balance. Member function debit should withdraw money from the Account and ensure that the debit amount does not exceed the Account's balance. If it does, the balance should be left unchanged, and the function should print a message indicating "Debit amount exceeded account balance." Member function getBalance should return the current balance. Use a main program that creates two Account objects and tests the member functions of class Account.

4. While exercising, you can use a heart-rate monitor to see that your heart rate stays within a safe range, as suggested by your trainers and doctors. According to the American Heart Association (AHA), the formula for calculating your maximum heart rate in beats per minute is 220 minus your age in years. Your target heart rate is in the range of 50–85% of your maximum heart rate.

   Create a class called HeartRates. The class attributes should include the person’s first name, last name, and birth year. Your class should have a constructor that receives this data as parameters. For each attribute, provide a set and get functions.

   The class also should include a function getAge that calculates and returns the person’s age (in years), a function getMaximumHeartRate that calculates and returns the person’s maximum heart rate, and a function getTargetHeartRate that calculates and returns the person’s target heart rate.

   Write an application that prompts for the person’s information, instantiates an object of class HeartRates and prints the information from that object—including the person’s first name, last name, and date of birth—then calculates and prints the person’s age in (years), maximum heart rate and target-heart-rate range.

5. Create a class called Time that will represent time in hours, minutes, and seconds. Include methods to add two times together and display the result (in hours, minutes, and seconds). Your program should accept input for time in seconds.