

CPP

Problem 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.

Problem 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.

Problem 3. Circle Class

Write a Circle class that has the following member variables:

- radius: a double
- pi: a double initialized with the value 3.14159

The class should have the following member functions:

- **Default Constructor.** A default constructor that sets radius to 0.0.
- **Constructor.** Accepts the radius of the circle as an argument.
- **setRadius.** A mutator function for the radius variable.
- **getRadius.** An accessor function for the radius variable.
- **getArea.** Returns the area of the circle, which is calculated as $\text{area} = \text{pi} * \text{radius} * \text{radius}$
- **getDiameter.** Returns the diameter of the circle, which is calculated as $\text{diameter} = \text{radius} * 2$
- **getCircumference.** Returns the circumference of the circle, which is calculated as $\text{circumference} = 2 * \text{pi} * \text{radius}$

Write a program that demonstrates the Circle class by asking the user for the circle's radius, creating a Circle object, and then reporting the circle's area, diameter, and circumference.

Problem 4. Freezing and Boiling Points

The following table lists the freezing and boiling points of several substances.

Substance	Freezing Point	Boiling Point
Ethyl Alcohol	-173	172
Oxygen	-362	-306

Design a class that stores a temperature in a temperature member variable and has the appropriate accessor and mutator functions. In addition to appropriate constructors, the class should have the following member functions:

- **isEthylFreezing**. This function should return the bool value true if the temperature stored in the temperature field is at or below the freezing point of ethyl alcohol. Otherwise, the function should return false.
- **isEthylBoiling**. This function should return the bool value true if the temperature stored in the temperature field is at or above the boiling point of ethyl alcohol. Otherwise, the function should return false.
- **isOxygenFreezing**. This function should return the bool value true if the temperature stored in the temperature field is at or below the freezing point of oxygen. Otherwise, the function should return false.
- **isOxygenBoiling**. This function should return the bool value true if the temperature stored in the temperature field is at or above the boiling point of oxygen. Otherwise, the function should return false.
- **isWaterFreezing**. This function should return the bool value true if the temperature stored in the temperature field is at or below the freezing point of water. Otherwise, the function should return false.
- **isWaterBoiling**. This function should return the bool value true if the temperature stored in the temperature field is at or above the boiling point of water. Otherwise, the function should return false.

Write a program that demonstrates the class. The program should ask the user to enter a temperature and then display a list of the substances that will freeze at that temperature and those that will boil at that temperature. For example, if the temperature is -20 the class should report that water will freeze and oxygen will boil at that temperature.

Problem 5. Month Class

Design a class named Month. The class should have the following private members:

- name A string object that holds the name of a month, such as "January," "February," etc.
- monthNumber An integer variable that holds the number of the month. For example, January would be 1, February would be 2, etc. Valid values for this variable are 1 through 12.

In addition, provide the following member functions:

- A default constructor that sets monthNumber to 1 and name to "January."

- A constructor that accepts the name of the month as an argument. It should set name to the value passed as the argument and set monthNumber to the correct value.
- A constructor that accepts the number of the month as an argument. It should set monthNumber to the value passed as the argument and set name to the correct month name.
- Appropriate set and get functions for the name and monthNumber member variables.
- Prefix and postfix overloaded ++ operator functions that increment monthNumber and set name to the name of next month. If monthNumber is set to 12 when these functions execute, they should set monthNumber to 1 and name to "January."
- Prefix and postfix overloaded -- operator functions that decrement monthNumber and set name to the name of previous month. If monthNumber is set to 1 when these functions execute, they should set monthNumber to 12 and name to "December."

Also, you should overload cout's << operator and cin's >> operator to work with the Month class. Demonstrate the class in a program.

Problem 6. PersonData and CustomerData classes

Design a class named PersonData with the following member variables:

- lastName
- firstName
- address
- city
- state
- zip
- phone

Write the appropriate accessor and mutator functions for these member variables. Next, design a class named CustomerData, which is derived from the PersonData class. The CustomerData class should have the following member variables:

- customerNumber
- mailingList

The customerNumber variable will be used to hold a unique integer for each customer. The mailingList variable should be a bool. It will be set to true if the customer wishes to be on a mailing list, or false if the customer does not wish to be on a mailing list. Write appropriate accessor and mutator functions for these member variables. Demonstrate an object of the CustomerData class in a simple program.