

Practice Problems for PE10

For **PE10**, you'll write both the header file and the implementation file for a **complete class**. You'll get partial credit for each part you complete; the header file will be graded separately from the implementation file. However, you can't have any syntax errors. Use the compile button to check that. Here are some additional ideas for problems you can practice with.



1 **TIMESPAN**

Define a class named `Timespan`. A `Timespan` stores a span of time in hours and minutes. The span between 8:00 and 10:30 am is 2 hours, 30 minutes. Hours is always non-negative while minutes is between 0 and 59 inclusive. You cannot have more than two `int` data members. (You may have less if you like.)

2 **CIRCLE**

Define a class named `Circle`. The class has two data members: a `Point` center and a double radius. The interface for the `Point` class has been provided.

3 **CONE**

Define a class named `Cone`, which represents a geometric shape that could be used in calculations. Each `Cone` object has two data members: radius and height.

4 CYLINDER

Define a class named `Cylinder`, which represents a geometric shape that could be used in calculations. Each `Cylinder` object has two data members: `radius` and `height`.

5 SPHERE

Define a class named `Sphere`. The class has two data members: a `Point` `center` and a `double` `radius`. The interface for the `Point` class has been provided.

6 POINT3D

Define a class named `Point3D`. The class has two three members: `x`, `y`, and `z`.

7 CALENDARDATE

Define a class named `CalendarDate`. The class has two three members: `day`, `month`, and `year`.

8 HISTORYLIST

Define a class named `HistoryList`. The class has two data members: `values` which is a `vector<int>` and `history`, which is a `vector<vector<int>>`. As modifications are made to the list, it is recorded in the history.

9 TICKET

Define a class named `Ticket` that represents a ticket to an on-campus event. A ticket has a price and also stores how many days early the ticket was bought. Some tickets have promotion codes that can allow special access and benefits to the ticket holder.

10 STUDENT

Define a class named `Student` that represents a college student. A student has a name, a year (1-4), and a `vector<Course>` representing courses the student is taking. The interface for the `Course` class is provided.

11 RIGGEDDICE

Define a class named `RiggedDice` that represents dice that let a player "cheat" by ensuring that they will always roll a value that is equal to or greater than a given minimum.

12 DAY

Define a class named `Day` that represents the day of the week.

13 BANKACCOUNT

Define a class named `BankAccount` that represents a bank account.

14 MOVIERATING

Define a class named `MovieRating` that keeps track of ratings for a movie.

15 POKEMON

Define a class named `Pokemon` that represents different kinds of Pokemon characters.

16 DIETER

Define a class named `Dieter` that models a person trying to monitor diet and weight to improve their Body Mass Index (BMI) rating.

17 CAR

Define a class named `Car` that represents an automobile. You will have data members representing the fuel efficiency (in miles-per-gallon) and the gallons of gas in the tank.

18 LINE

Define a class named `Line` that represents a 2D line. The class has two data members for the end points that are of type `Point`. The interface for the `Point` class has been provided.

19 RECTANGLE

Define a class named `Rectangle` that represents a rectangular two-dimensional region.

20 CLOCKTIME

Define a class named `ClockTime` that has been defined for storing information about times of day. Each `ClockTime` object keeps track of hours, minutes and a string to indicate "am" or "pm".

21 PRODUCT

Define a class named `Product`. The `Product` class represents an item that would appear for sale on a Web site or in a department store (like a toaster). Each `Product` object has three data members: `description`, `productNumber` and `price`.

22 COMPLEX

Define a class named `Complex` that stores complex numbers. Complex numbers are those that can be written as $(x + yi)$ where x and y are real numbers and i is the square root of -1 . In other words, complex numbers have a real part (x) and an imaginary part (y).

23 DATE

Define a class named `Date` to store information about calendar dates. Each `Date` has a month, day and year.

24 POINT

Define a class named `Point` that represents a 2D point with x and y coordinates (real numbers).

25 RADIOSTATION

Define a class named `RadioStation` that has been defined for storing information about radio stations. Each has a name (string), its broadcast band (string) and its station number (a real number).

26 USCURRENCY

Define a class named `USCurrency` that represents currency amounts in dollars and cents (both integers) where one dollar equals 100 cents.