Consider the class Point in Program 1. The addition of two points is done by adding the coordinates x and y of the points accordingly. For example, if the points are (1,2) and (3,4), thus adding the points results in a new point, (4, 6). The addition operation can be implemented in several ways, such as using operators or functions. Answer questions (a) to (c) below.

1. Define an overloaded operator for the class that performs point additions.
2. Define a regular function that performs the addition of two points.
3. In the main function, using the operator and function from (a) and (b), write two different ways to add the points p and q and store the result into r.

//Program 1

#include <iostream>

using namespace std;

class Point{

private:

int x, y;

public:

Point(int \_x=0, int \_y=0) { x = \_x; y = \_y; }

int getX() const { return x; }

int getY() const { return y; }

void input() { cin >> x >> y; }

// Answer for question (1a)

}; // end of class Point

// Answer for question (1b)

int main(){

Point p, q, r;

p.input();

q.input();

// Answer for question (1c)

return 0;

}// end of main

Complete Program 2 which asks the user to enter a word, then counts the number of capital letters in it and creates a new word in a reversed form. Figure below shows an example run and expected output of the program. The bold text indicates user input.

Enter a word: **WelComE**

Number of capital letters: 3

The reversed word: EmoCleW

// Program 2

#include <iostream>

#include <string>

#include <cctype>

using namespace std;

int main(){

string w; // user input

string r; // reversed word

int cap; // number of capital letters

cout << "Enter a word: ";

cin >> w;

// Answer for question

cout << "Number of capital letters: " << cap << endl;

cout << "The reversed word: " << r << endl;

return 0;

}// end of main

A rectangle can be represented by its bottom-left corner point (x1, y1) and top-right corner point (x2, y2), as shown in Figure below. The width and height are then determined from these coordinates.

Shape, rectangle

Description automatically generated

Complete Program 3 below based on the tasks given. Write your answer in the program.

// Program 3

#include<iostream>

using namespace std;

class Rectangle{

private:

int x1,y1; // bottom-left corner

int x2,y2; // top-right corner

public:

Rectangle(){ x1=y1=x2=y2=0; } // The default constructor

Rectangle(int a, int b, int c, int d){ // The constructor that sets the bottom-left and top-right corners

x1=a; y1=b;

x2=c; y2=d;

}

// Define a constructor to create a rectangle with the bottom-left corner (0,0)

// and the width and height set as w and h, respectively.

// Define a constructor to create a square with the bottom-left corner (x, y), and the side length of s.

// Define a member function (method) that determines the width of the rectangle.

// Define a member function (method) that determines the height of the rectangle.

friend void inputRectangle(Rectangle&);

}; // End of class

// Write the definition code for the function inputRectangle to read the coordinates of a rectangle from the keyboard.

int main(){

// Declare an array to hold 4 objects of Rectangle, and initialize the array as follow:

// the first element is a rectangle with bottom-left and top-right corners of (1,2) and (3,4), respectively.

// the second element is a rectangle with bottom-left corner at (0,0), and width is 5 and height is 10 .

// the third element is a square with bottom-left corner at (3,4), and the side length is 5

// Read the data for the fourth element of the array from the keyboard.

// Using a loop, calculate the total area of the rectangles and print the result.

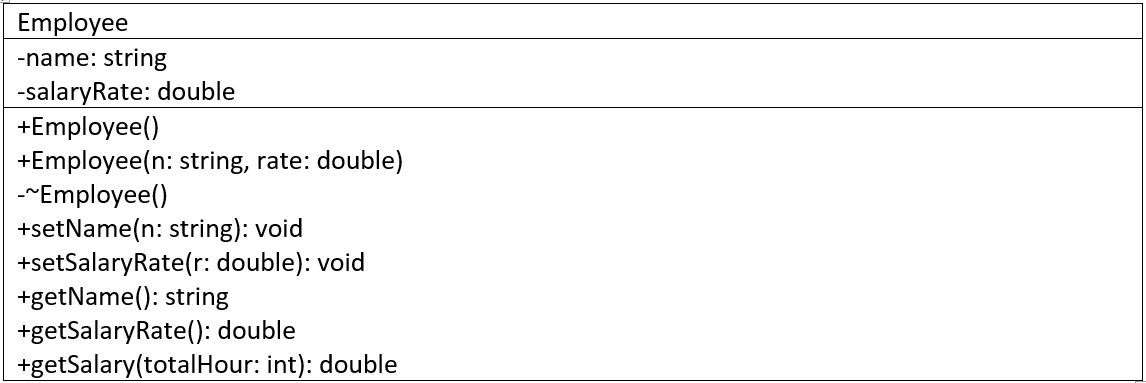
return 0;

}

Design a UML class diagram to model an employee of a company. Each employee has a name and salary rate (given as per hour). You must use proper notations for each element of the design including notations for private and public members and data types for each member data (attribute) and parameter. The class should also include the following operations:

* A default constructor.
* An appropriate overloaded constructor.
* A destructor.
* A mutator for each attribute.
* An accessor for each attribute.
* A method to calculate the total salary earned. The method will take the total hours of work as a parameter and return the total salary.

Answer



Define a constructor to create a record for employee with the name and salary rate set as n and rate, respectively.

Answer:

Employee ::Employee (){

name= “Ieskandar Zulqarnain”; rate = 9.50 ;

}

Employee::Employee(string n, double sr){

name= n; rate= sr;

}

Define a member function (method) that calculates the total hours of work and total amount of salary earned.

Answer:

double Employee::Totsalary (int hour)

return rate\*hour;