

PS # 10

Deadline: 07.06.2020 at 23:50

Hand-in Policy:

Create a cpp file for each part of the assignment and save the screenshot while these parts are running.

Your assignment should consist of 3 cpp files and 3 screenshots.

Zip all files of the assignment. Your zip files should look like this

Ps10_studentnumber_name_surname.zip

1)

Create a base class called Vehicle that has the manufacturer's name (type string), number of cylinders in the engine (type int), and owner (type Person given in the code that follows). Then create a class called Truck that is derived from Vehicle and has additional properties, the load capacity in tons (type double since it may contain a fractional part) and towing capacity in pounds (type int). Be sure your classes have a reasonable complement of constructors and accessor methods, an overloaded assignment operator, and a copy constructor. Write a driver program that tests all your methods. The definition of the class Person follows. The implementation of the class is part of this programming project.

class Person

{

public:

Person();

Person(string theName);

Person(const Person& theObject);

string getName() const;

Person& operator=(const Person& rtSide);

friend istream& operator >>(istream& inStream,

Person& personObject);

friend ostream& operator <<(ostream& outStream,

const Person& personObject);

private:

string name;

};

2)

Define a class named `Payment` that contains a member variable of type `float` that stores the amount of the payment and appropriate accessor and mutator functions. Also create a member function named `paymentDetails` that outputs an English sentence describing the amount of the payment.

Next, define a class named `CashPayment` that is derived from `Payment`. This class should redefine the `paymentDetails` function to indicate that the payment is in cash. Include appropriate constructor(s). Define a class named `CreditCardPayment` that is derived from `Payment`. This class should contain member variables for the name on the card, expiration date, and credit card number. Include appropriate constructor(s). Finally, redefine the `paymentDetails` function to include all credit card information in the printout. Create a main function that creates at least two `CashPayment` and two `CreditCardPayment` objects with different values and calls to `paymentDetails` for each.

3)

Define a class named `Document` that contains a member variable of type `string` named `text` that stores any textual content for the document. Create a function named `getText` that returns the text field, a way to set this value, and an overloaded assignment operator.

Next, define a class for `Email` that is derived from `Document` and that includes member variables for the sender, recipient, and title of an e-mail message. Implement appropriate accessor and mutator functions. The body of the e-mail message should be stored in the inherited variable `text`. Also overload the assignment operator for this class.

Similarly, define a class for `File` that is derived from `Document` and that includes a member variable for the pathname. Implement appropriate accessor and mutator functions for the pathname and overload the assignment operator.

Finally, create several sample objects of type `Email` and `File` in your main function. Test your objects by passing them to the following subroutine, which will return `true` if the object contains the specified keyword in the text property.

```
bool ContainsKeyword( const Document& docObject, string keyword)
{
    if (docObject.getText().find(keyword) != string::npos)
        return true;
    return false;
}
```

For example, you might test to see whether an e-mail message contains the text `"c++"` with the call `ContainsKeyword(emailObj, "c++");`.