PreferredCustomer Class

Stephen Gerkin
September 25, 2019
CIST 2362 – C++ II
CRN 22282
Programming Lab 06

Program Description

PreferredCustomer Class demonstrates inheritance using a base class of **PersonData** containing member variables for information about a person and a derived class **CustomerData** for storing information about a customer. **PreferredCustomer** is derived from **CustomerData** and a basic program is used to demonstrate the functionality of the methods unique to **PreferredCustomer** by creating a new object and then allowing the user to set the amount of purchases. A discount level is then determined and displayed to the user.

Class Documentation

PersonData Class Reference

PersonData contains data for a Person.

Public Member Functions

- **PersonData** ()
  No argument constructor initializes all member variables to empty strings.

- **PersonData** (const std::string &lastName, const std::string &firstName, const std::string &address, const std::string &city, const std::string &state, const std::string &zip, const std::string &phone)
  Full argument constructor.

- **PersonData** (const **PersonData** &rhs)=default
- **PersonData** & **operator=** (const **PersonData** &rhs)
  Copy assignment operator overload.

- **PersonData** (**PersonData** &&rhs) noexcept
  Move constructor.

- **PersonData** & **operator=** (**PersonData** *rhs) noexcept
  Move assignment overload.

- void **setLastName** (const std::string &lastName)
  Setter for lastName.

- const std::string & **getFirstName** () const
  Getter for firstName.

- void **setFirstName** (const std::string &firstName)
  Setter for firstName.

- const std::string & **getAddress** () const
  Getter for address.

- void **setAddress** (const std::string &address)
  Setter for address.

- const std::string & **getCity** () const
  Getter for city.

- void **setCity** (const std::string &city)
  Setter for city.

- const std::string & **getState** () const
  Getter for state.

- void **setState** (const std::string &state)
  Setter for state.

- const std::string & **getZip** () const
  Getter for zip.

- void **setZip** (const std::string &zip)
  Setter for zip.

- const std::string & **getPhone** () const
  Getter for phone.

- void **setPhone** (const std::string &phone)
  Setter for phone.

- const std::string & **getLastName** () const
  Getter for lastName.

Protected Attributes

- std::string **lastName**
- std::string **firstName**
- std::string **address**
- std::string **city**
- std::string **state**
- std::string **zip**
- std::string **phone**

Detailed Description

PersonData contains data for a Person. Provides getters and setters for each field. All fields are strings for simplicity and require outside conversion if necessary. This is to allow multiple formats for fields such as State that could be in either the form 'GA' or 'Georgia'. Additionally, this allows entry of zip codes that are full form (99999-9999) or simple form (99999).

Constructor & Destructor Documentation

PersonData::PersonData (const std::string & *lastName*, const std::string & *firstName*, const std::string & *address*, const std::string & *city*, const std::string & *state*, const std::string & *zip*, const std::string & *phone*)

Full argument constructor.

Parameters

lastName	Person's last name.
firstName	Person's first name.
address	Person's address.
city	Person's city.
state	Person's state.
zip	Person's zipcode.
phone	Person's phone number.

PersonData::PersonData (PersonData && *rhs*) `[noexcept]`

Move constructor.

Parameters

rhs	Object to move into new constructed object.

Member Function Documentation

const std::string & PersonData::getAddress () const

Getter for address.

Returns

address.

const std::string & PersonData::getCity () const

Getter for city.

Returns

city.

const std::string & PersonData::getFirstName () const

Getter for firstName.

Returns

firstName.

const std::string & PersonData::getLastName () const

Getter for lastName.

Returns

lastName.

const std::string & PersonData::getPhone () const

Getter for phone.

Returns

phone.

const std::string & PersonData::getState () const

Getter for state.

Returns

state.

const std::string & PersonData::getZip () const

Getter for zip.

Returns

zip.

PersonData & PersonData::operator= (const PersonData & *rhs*)

Copy assignment operator overload.

Parameters

rhs	Object to copy

Returns

*this as copy of the object.

PersonData & PersonData::operator= (PersonData * *rhs*) **[noexcept]**

Move assignment overload.

Parameters

rhs	Object to move into *this.

Returns

*this with new, moved data.

void PersonData::setAddress (const std::string & *address*)

Setter for address.

Parameters

address	new string to set address to.

void PersonData::setCity (const std::string & *city*)

Setter for city.

Parameters

city	new string to set city to.

void PersonData::setFirstName (const std::string & *firstName*)

Setter for firstName.

Parameters

firstName	new string to set firstName to.

void PersonData::setLastName (const std::string & *lastName*)

Setter for lastName.

Parameters

lastName	new string to set lastName to.

void PersonData::setPhone (const std::string & *phone*)

Setter for phone.

Parameters

phone	new string to set phone to.

void PersonData::setState (const std::string & *state*)

Setter for state.

Parameters

state	new string to set state to.

void PersonData::setZip (const std::string & *zip*)

Setter for zip.

Parameters

zip	new string to set zip to.

CustomerData Class Reference

CustomerData contains information about a customer.

Public Member Functions

- **CustomerData** (const std::string &lastName, const std::string &firstName, const std::string &address, const std::string &city, const std::string &state, const std::string &zip, const std::string &phone, unsigned int customerNumber, bool mailingList)
  Full argument constructor.

- **CustomerData** (const **CustomerData** &rhs)=default
- **CustomerData** & **operator=** (const **CustomerData** &rhs)
  Copy assignment overload. Creates a new object by calling the base class constructor and setting the member variables.

- **CustomerData** (**CustomerData** &&rhs) noexcept=default
- **CustomerData** & **operator=** (**CustomerData** &&rhs) noexcept=default
- unsigned int **getCustomerNumber** () const
  Getter for customerNumber.

- void **setCustomerNumber** (unsigned int customerNumber)
  Setter for customerNumber.

- bool **isMailingList** () const
  Getter for mailingList.

- void **setMailingList** (bool mailingList)
  Setter for mailing list.

Detailed Description

CustomerData contains information about a customer.

Derived from **PersonData** to contain information about a person. Has member variables for storing a unique customer ID number and a boolean for determining if the customer should receiving mailing list items. Provides getters and setters for all variables.

Constructor & Destructor Documentation

CustomerData::CustomerData (const std::string & *lastName*, const std::string & *firstName*, const std::string & *address*, const std::string & *city*, const std::string & *state*, const std::string & *zip*, const std::string & *phone*, unsigned int *customerNumber*, bool *mailingList*)

Full argument constructor.

Parameters

lastName	Customer's last name.
firstName	Customer's first name.
address	Customer's address.
city	Customer's city.
state	Customer's state.
zip	Customer's zip code.
phone	Customer's phone.
customerNumber	Customer's unique ID number.
mailingList	Boolean value for if customer is on mailing list.

Member Function Documentation

unsigned int CustomerData::getCustomerNumber () const

Getter for customerNumber.

Returns

customerNumber.

bool CustomerData::isMailingList () const

Getter for mailingList.

Returns

true if on mailing list.

CustomerData & CustomerData::operator= (const CustomerData & *rhs*)

Copy assignment overload. Creates a new object by calling the base class constructor and setting the member variables.

Parameters

rhs	The object to copy.

Returns

*this with copied values.

void CustomerData::setCustomerNumber (unsigned int *customerNumber*)

Setter for customerNumber.

Parameters

customerNumber	value to set customerNumber to.

void CustomerData::setMailingList (bool *mailingList*)

Setter for mailing list.

Parameters

mailingList	true if customer is on mailing list.

PreferredCustomer Class Reference

PreferredCustomer contains information about a Preferred Customer. Provides a getter/setter for total purchased amount and a method to get the discount level for the customer.

Public Member Functions

- **PreferredCustomer** ()
  No argument constructor. Sets member variables to 0.

- **PreferredCustomer** (const **CustomerData** &obj)

Copy constructor. Sets discount level on creation in case bad data was provided for original object's discount level (this should never happen but this provides additional safety).

- **PreferredCustomer** (const **CustomerData** &obj, double purchasesAmount)
  *Constructor for converting a **CustomerData** object into a Preferred Customer object. Sets discount level after initializing with provided purchasesAmount.*

- **PreferredCustomer** (const std::string &lastName, const std::string &firstName, const std::string &address, const std::string &city, const std::string &state, const std::string &zip, const std::string &phone, unsigned int customerNumber, bool mailingList, double purchasesAmount)
  Full argument constructor. Sets discount level after initializing all member variables and base class variables.

- **PreferredCustomer** (const **PreferredCustomer** &rhs)=default
- **PreferredCustomer** & **operator=** (const **PreferredCustomer** &rhs)=default
- **PreferredCustomer** (**PreferredCustomer** &&rhs) noexcept=default
- **PreferredCustomer** & **operator=** (**PreferredCustomer** &&rhs) noexcept=default
- double **getPurchasesAmount** () const
  Getter for purchasesAmount.

- void **setPurchasesAmount** (double amount)
  setter for purchasesAmount

- void **addPurchase** (double amount)
- void **removePurchase** (double amount)
- double **getDiscountLevel** () const
  Getter for discountLevel.

Detailed Description

PreferredCustomer contains information about a Preferred Customer. Provides a getter/setter for total purchased amount and a method to get the discount level for the customer.

Constructor & Destructor Documentation

PreferredCustomer::PreferredCustomer (const CustomerData & *obj*) `[explicit]`

Copy constructor. Sets discount level on creation in case bad data was provided for original object's discount level (this should never happen but this provides additional safety).

Parameters

obj	object to use for construction.

PreferredCustomer::PreferredCustomer (const CustomerData & *obj*, double *purchasesAmount*)

Constructor for converting a **CustomerData** object into a Preferred Customer object. Sets discount level after initializing with provided purchasesAmount.

Parameters

obj	**CustomerData** object to create new object from.
purchasesAmount	value to initialize purchasesAmount with.

PreferredCustomer::PreferredCustomer (const std::string & *lastName*, const std::string & *firstName*, const std::string & *address*, const std::string & *city*, const std::string & *state*, const std::string & *zip*, const std::string & *phone*, unsigned int *customerNumber*, bool *mailingList*, double *purchasesAmount*)

Full argument constructor. Sets discount level after initializing all member variables and base class variables.

Parameters

lastName	Customer's last name.
firstName	Customer's first name.
address	Customer's address.
city	Customer's city.
state	Customer's state.
zip	Customer's zipcode.
phone	Customer's phone number.
customerNumber	Customer's unique ID number.
mailingList	Boolean value for if the customer is on the mailing list.
purchasesAmount	value to initialize purchasesAmount with.

Member Function Documentation

double PreferredCustomer::getDiscountLevel () const

Getter for discountLevel.

Returns

decimal value for percent discount.

double PreferredCustomer::getPurchasesAmount () const

Getter for purchasesAmount.

Returns

purchasesAmount.

void PreferredCustomer::setPurchasesAmount (double *amount*) `[noexcept]`

setter for purchasesAmount

Parameters

amount	amount to set purchasesAmount to.

Exceptions

NegativePurchase	exception if amount parameter is a negative value.

NegativePurchase Class Reference

Exception object for negative values entered in a purchase amount.

Public Member Functions

- **NegativePurchase** (double value)
  Default constructor.

- double **getValue** () const
  Getter for value.

Detailed Description

Exception object for negative values entered in a purchase amount. Contains the value that was used.

Constructor & Destructor Documentation

NegativePurchase::NegativePurchase (double *value*) `[explicit]`

Default constructor.

Parameters

value	Exceptional value used that is not valid.

Member Function Documentation

double NegativePurchase::getValue () const

Getter for value.

Returns

value.

File Documentation

CustomerDataDemo.cpp File Reference

```
#include <string>
#include <iostream>
#include <iomanip>
#include <regex>
#include "CustomerDataDemo.h"
#include "NegativePurchase.h"
```

Functions

- void **startDemo** ()
  *Starts the demonstration of the **PreferredCustomer** class.*

- void **constructCustomer** (**PreferredCustomer** &customer)
  *Constructs a **PreferredCustomer** object from user input.*

- std::string **getStringField** (const std::string &fieldName)
  Gets a string for a specified field from the user.

- int **getCustomerNumber** ()
  Gets an integer value from the user to use as a customer ID number.

- bool **getMailingListStatus** ()
  Gets a boolean value from the user determining if a customer is on the mailing list.

- void **modifySales** (**PreferredCustomer** &customer)
  *Gets a sales amount from the user to modify the **PreferredCustomer** object with.*

- void **displayPurchasesAmount** (const **PreferredCustomer** &customer)
  Displays the current sales amount and discount level for the customer.

- bool **checkModifySalesAgain** ()
  Determines if the user wants to modify the sales amount again.

Function Documentation

bool checkModifySalesAgain ()

Determines if the user wants to modify the sales amount again.

Returns

True if the user wants to continue modifying sales amount.

void constructCustomer (PreferredCustomer & *customer*)

Constructs a **PreferredCustomer** object from user input.

Parameters

customer	The object to construct.

void displayPurchasesAmount (const PreferredCustomer & *customer*)

Displays the current sales amount and discount level for the customer.

Parameters

customer	The object we want to display the information for.

int getCustomerNumber ()

Gets an integer value from the user to use as a customer ID number.

Returns

User input as integer.

bool getMailingListStatus ()

Gets a boolean value from the user determining if a customer is on the mailing list.

Returns

User input as boolean.

std::string getStringField (const std::string & *fieldName*)

Gets a string for a specified field from the user.

Parameters

fieldName	The field the user should enter data for.

Returns

The user input as a string.

void modifySales (PreferredCustomer & *customer*)

Gets a sales amount from the user to modify the **PreferredCustomer** object with. Provides functionality to show the current sales and discount level.

Parameters

customer	The customer object we're working on.

void startDemo ()

Starts the demonstration of the **PreferredCustomer** class. Creates a new class and has the user modify the sales purchase amounts to display the discount levels. Continues to allow modifying of sales while user desires.

main.cpp File Reference

```
#include <iostream>
#include <string>
#include <regex>
```

Functions

- void **startDemo** ()
  *Starts the demonstration of the **PreferredCustomer** class.*

- bool **repeatProgram** ()
  Function to repeat program.

- int **main** ()
  Main method entry point for program.

Function Documentation

int main ()

Main method entry point for program.

bool repeatProgram ()

Function to repeat program. Displays prompt to user asking if they would like to repeat the program. Waits for input consisting of "y" or "n" (case insensitive).

Returns

True if user wants to repeat program.

void startDemo ()

Starts the demonstration of the **PreferredCustomer** class. Creates a new class and has the user modify the sales purchase amounts to display the discount levels. Continues to allow modifying of sales while user desires.

Sample Program Output