# **Customer**

### Constructors

// Customer

//desc: Constructs a Customer object with empty string values

**Customer()**

// Customer

//desc: Constructs a Customer object

**Customer(Customer c)**

// Customer

//params: id, first name, last name, middle name, maiden last name, phone, suffix and address Strings

// that are not modified in the constructor

//desc: Constructs a Customer object with the argument values

**Customer(int id, String fn, String ln, String mn, String mln, String phone, String suffix, String  
 address)**

Initializes the Customer object with argument values.

### Accessors

//getFirstName

//desc: Returns the customer first name variable

//return: String

**String getFirstName()**

//getLastName

//desc: Returns the customer last name variable

//return: String

**String getLastName()**

//getPhone

//desc: Returns the customer phone number variable.

//return: String

**String getPhone()**

**//**getMiddleName

//desc: Returns the customer middle name variable.

//return: String

**String getMiddleName()**

//getMaternalLastName

//desc: Returns the customer maternal last name variable.

//return: String  
**String getMaternalLastName()**

//getPrefix

//desc: Returns the customer prefix variable.

//return: String

**String getPrefix()**

//getSuffix

//desc: Returns the customer suffix variable.

//return: String

**String getSuffix()**.

//getAddress

//desc: Returns the customer address variable.

//return: String

**String getAddress()**

### Mutators

//setFirstName

//params: String value that represents the customer first name

//desc: Modifies the customer first name variable with the given argument

**void setFirstName(String fn)**

//setLastName

//params: String value that represents the customer last name

//desc: Modifies the customer last name variable with the given argument

**void setLastName(String ln)**

//setPhone

//params: String value that represents the customer phone number

//desc: Modifies the customer phone variable with the given argument

**void setPhone(String phn)**

//setMiddleName

//params: String value that represents the customer middle name

//desc: Modifies the customer middle name variable with the given argument

**void setMiddleName(String mn)**

//setMaternalLastName

//params: String value that represents the customer maternal last name

//desc: Modifies the customer maternal last name variable with the given argument

**void setMaternalLastName(String mln)**

//setPrefix

//params: String value that represents the customer prefix

//desc: Modifies the customer prefix variable with the given argument

**void setPrefix(String pre)**

//setSuffix

//params: String value that represents the customer suffix

//desc: Modifies the customer suffix variable with the given argument

**void setSuffix(String sfx)**

//setAddress

//params: String value that represents the customer address

//desc: Modifies the customer address variable with the given argument.

**void setAddress(String addr)**

### Miscellaneous

**//**toString

//desc: Returns a string that contains all information about a customer.

//return: String

**String toString()**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **DataHandler**

### Constructors

//DataHandler

//desc: Constructs a DataHandler object with empty DB connection String

**DataHandler()**

**//**SetDatabaseConnectionString

//params: String argument representing the database connection String

//desc: Modifies the database connection string variable with the given argument

**void SetDatabaseConnectionString(String dbcs)**

### Query Functions

//retrieveCustomer

//params: Integer that represents the Customer ID to be retrieved

//desc: Queries the database for a customer and returns it if found, and null if not

// return: Customer object

**Customer queryCustomer(int CustomerID)**

//retrieveInvoice

//params: Integer that represents the Invoice ID to be retrieved

//desc: Queries the database for an Invoice and returns it if found, and null if not

// return: Invoice object

**Invoice queryInvoice(int InvoiceID)**

//retrieveCheck

//params: Integer that represents the Check ID to be retrieved

//desc: Queries the database for a Check and returns it if found, and null if not

// return: Check object

**Check queryCheck(int checkID)**

//retrieveUser

//params: Integer that represents the User ID to be retrieved

//desc: Queries the database for a User and returns it if found, and null if not

// return: User object

**User queryUser(int userID)**

### Customer Handler

//updateCustomer

//params: Customer object

//desc: Queries and updates the customer from the database

// Returns true if successful and false if not

//return: Boolean

**boolean updateCustomer(Customer c)**

//createCustomer

//params: Customer object

//desc: Inserts a new customer to the database with argument customer data

// Returns true if successful and false if not

//return:Boolean

**boolean ceateCustomer(Customer c)**

//removeCustomer

//params: Customer object representing the customer to be removed from database

//desc: Deletes a customer from the database using the argument object

// Returns true if successful and false if not

//return: Boolean

**boolean removeCustomer(Customer c)**

### Invoice Handler

//updateInvoice

//params: Invoice object

//desc: Queries and updates the invoice from the database

// Returns true if successful and false if not

//return: Boolean

**boolean updateInvoice(Invoice i)**

//createInvoice

//params: Invoice object

//desc: Inserts a new Invoice to the database with argument customer data

// Returns true if successful and false if not

//return: Boolean

**boolean createInvoice(Invoice i)**

//removeInvoice

//params: Customer object representing the invoice to be removed from database

//desc: Deletes an invoice from the database using the argument object

// Returns true if successful and false if not

//return: Boolean

**boolean removeInvoice(Invoice i)**

### Check Handler

//updateCheck

//params: Check object

//desc: Queries and updates the Check from the database

// Returns true if successful and false if not

//return: Boolean

**boolean updateCheck(Check c)**

//createCheck

//params: Check object

//desc: Inserts a new Check to the database with argument customer data

// Returns true if successful and false if not

//return:Boolean

**boolean createCheck(Check c)**

//removeCheck

//params: Check object representing the customer to be removed from database

//desc: Deletes a check from the database using the argument object

// Returns true if successful and false if not

//return: Boolean

**boolean removeCheck(Check c)**

### Miscellaneous

//getAllCustomers

//desc: Queries database for all customers and stores in a Customer array.

//return: Customer object array

**Customer[ ] getAllCustomers()**

//getAlInvoices

//desc: Queries database for all invoices and stores in an Invoice array.

//return: Invoice object array

**Invoice[ ] getAllInvoices()**

//getAllChecks

//desc: Queries database for all checks and stores in a Check array

//return: Check object array

**Check[ ] getAllChecks()**.

//toString

//desc: Returns a string that contains information about the DataHandler class

//return: String

**String toString()**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Check**

## Constructors

// Check

//desc: Constructs a Check object with empty values

**Check()**

// Check  
//params: Check object representing the check to be copied

//desc: Constructs a Check object from argument Check

**Check(Check c)**

// Check

//params: check amount, routing number, check number, check date

//desc: Constructs a Check object with argument values

**Check(float amt, int rn, int cn, Date cd)**

## Accessors

//getAmount

//desc: Returns the check amount value

//return: float

**float getAmount()**

//getCheckNumber

//desc: Returns the check number value

//return: int

**int getCheckNumber()**

//getRoutingNumber

//desc: Returns the routing number value

//return: int

**int getRoutingNumber()**

//getCheckDate

//desc: Returns the date of the check

//return: Date object

**Date getCheckDate()**

//getCompanyName

//desc: Returns the value of the company name

//return: String

**String getCompanyName()**

Returns the company name variable.

## Mutators

//setAmount

//params: float value representing the new amount of the check

//desc: Modifies the check amount variable with the argument variable

**void setAmount(float amt)**

//setCheckNumber

//params: int value representing the new check number variable

//desc: Modifies the check number variable with the argument variable

**void setCheckNumber(int cn)**

//setRoutingNumber

//params: int value representing the new routing number variable

//desc: Modifies the check routing number variable with the argument variable

**void setRoutingNumber(int rn)**

//setCheckDate

//params: Date object representing the new check date variable

//desc: Modifies the check date variable with the argument variable

**void setCheckDate(Date d)**

//setCompanyName

//params: String value representing the new company name

//desc: Modifies the company name variable with the argument variable

**void setCompanyName(String co)**

## Miscellaneous

//equals

//params: Check object that will be compared with the current object

//desc: returns true if current object and argument object are equal, false if not

//return: boolean

**boolean equals(Check c)**

//toString

//desc: returns a string containing all the information of a check

//return: String

**String toString()**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Invoice**

## Constructors

//Invoice

//desc: Constructs an Invoice object with empty values

**Invoice()**

//Invoice  
//params: Invoice object representing the invoice to be copied

//desc: Constructs an Invoice object from the argument Invoice object

**Invoice(Invoice inv)**

//Invoice

//params: Date of invoice, array of checks, customer, user

//desc: Constructs an Invoice object from argument values

**Invoice(Date id, Check[] chkArr, Customer c, User u)**

## Accessors

//getInvoiceDate

//desc: Returns the date of this invoice

//return: Date object

**Date getInvoiceDate()**

//getChecksArray

//desc: returns the array object containing all checks for this invoice

//return: Check array object

**Check[ ] getChecksArray()**

//getCustomer

//desc: returns the Customer associated with this invoice

//return: Customer object

**Customer getCustomer()**

//getUser

//desc: returns the user associated with this invoice

//return: User object

**User getUser()**

## Mutators

//setInvoiceDate

//params: Date objet representing the new invoice date

//desc: Modifies the invoice date variable with the given argument

**setInvoiceDate(Date invDate)**

//setChecksArray

//params: Check array object representing the new checks

//desc: Modifies the checks array with the given argument.

**setChecksArray(Check[] chkArr)**

//setCustomer

//params: Customer object representing the new customer

//desc: Modifies the Customer object with the given argument

**setCustomer(Customer cust)**

//setUser

//params: User object representing the new user

//desc: Modifies the User object with the given argument

**setUser(User u)**

## Miscellaneous

//equals

//params: Invoice object representing the invoice to be compared to

//desc: Returns true if the argument Invoice object is equal to the current Invoice object

//return: Boolean

**boolean equals(Invoice inv)**

//toString

//desc: Returns a string that contains information about the Invoice class.

//return: String

**String toString()**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **User**

## Constructors

**//**User

//desc: Constructs a User object with default values

**User()**

//User

//params: User object representing the User to be copied

//desc: Constructs a User object form the argument User

**User(User u)**

//User

//params: String username, String password

//desc: Constructs a user with the argument values

**User(String un, String pwd)**

## Accessors

//getUsername

//desc: returns the value of the username

//return: String

**String getUsername()**

//getPassword

//desc: returns the value of the User password

//return: String

**String getPassword()**

## Mutators

//setUsername

//params: String representing the new username for the user

//desc: Modifies the username variable with the given argument value

**void setUsername(String un)**

//setPassword

//params: String representing the new password for the user

//desc: Modifies the password variable using the argument value

**void setPassword(String pwd)**

## Miscellaneous

//equals

//params: User object that will be compared to the current object

//desc: Returns true if the argument User object is equal to the current User object

//return: boolean

**boolean equals(User u)**

//toString

//desc: Returns a string that contains information about the User class.

//return: String

**String toString()**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_