**Team Name:**

The Coding Company

**Team Members:**

Daniel Martinez

Francisco Fierro

Kyle Pamintuan

Matthew Le

**Team Leader:**

Matthew Le

Account

* getUserID()

**Method:** getUserID()

**Description:** Returns the User’s ID, a unique identifier number assigned to each account

**PreConditions:** N/A

**PostConditions:** The User’s ID is returned

**Signature:** String getUserID();

* getUserName()

**Method:** getUserName()

**Description:** Returns user’s username which is displayed in the chatroom

**PreConditions:** N/A

**PostConditions:** The user’s username is returned to the control object.

**Signature:** String getUserName();

* setUserName()

**Method:** setUserName()

**Description:** Sets the username of this Account object instance

**PreConditions:** The username being set must not contain explicit language, must be at least 6 characters in length, and must be unique.

**PostConditions:** N/A

**Signature:** void setUserName(String name);

* getUserType()

**Method:** getUserType()

**Description:** Returns whether this Account object is a Student or Instructor

**PreConditions:** N/A

**PostConditions:** Returns true if the account is an Instructor or false if the account is a student

**Signature:** Boolean getUserType();

* getRepPoints()

**Method:** getRepPoints()

**Description:** Returns this Account object’s reputation points

**PreConditions:** N/A

**PostConditions:** The user’s reputation points are returned to the control object

**Signature:** void getRepPoints();

* setRepPoints()

**Method:** setRepPoints()

**Description:** Sets this Account instance’s reputation points

**PreConditions:** The argument should be a signed integer that represents the displacement of the user’s reputation points ie the points by which the user’s reputation will increase/decrease

**PostConditions:** The object’s points attribute is updated

**Signature:** void setRepPoints(int points);

* getName()

**Method:** getName()

**Description:** Gets the name of the user

**PreConditions:** N/A

**PostConditions:** The student’s name is returned

**Signature:** String getName();

Chatroom

* getAccessCode()

**Method:** getAccessCode()

**Description:** Return the access code

**PreConditions:** N/A

**PostConditions:** The chatroom’s access code must be returned

**Signature:** String getAccessCode()

* setAccessCode()

**Method:** setAccessCode()

**Description:** Edit the access code of a chatroom and save the changes.

**PreConditions:** Access code must not currently be in use by another chatroom

**PostConditions:** Access code has been edited.

**Signature:** void setAccessCode(String code)

* getAttendance()

**Method:** getAttendance()

**Description:** Returns list of users that have successfully accessed the chatroom.

**PreConditions:**The account calling must be an instructor account

**PostConditions:** List of students is returned to the control object.

**Signature:** String getAttendance()

* setAttendance()

**Method:** setAttendance()

**Description:** Edit the list of students who have accessed the chatroom.

**PreConditions:** No new changes to the attendance roster.

**PostConditions:** List of attendance roster has been edited and saved.

**Signature:** void setAttendance(string arr[])

* getComment()

**Method:** getComment()

**Description:** Returns a comment object in the chatroom

**PreConditions:** The index argument must be valid

**PostConditions:** A comment object is returned

**Signature:** Comment getComment(int index)

* addStudent():

**Method:** addStudent()

**Description:** Returns the timestamp of a Notification object

**PreConditions:** The account with userID must not currently be in chatroom

**PostConditions:** N/A

**Signature:** void addStudent(String userID)

* removeStudent():

**Method:** removeStudent()

**Description:** Removes student from the chatroom

**PreConditions:** The account with userID must currently be in chatroom

**PostConditions:** The student object is removed from the chatroom

**Signature:** void removeStudent(String userID)

* addComment()

**Method:** addComment()

**Description:** Add a comment into the chatroom

**PreConditions:** Comment must meet requirements

**PostConditions:** A comment was added into the chatroom

**Signature:** void addComment(Comment comm)

* removeComment()

**Method:** removeComment()

**Description:** Remove a comment from the chatroom

**PreConditions:** The given index must be valid

**PostConditions:** The comment was removed from the chatroom

**Signature:** void removeComment(int index)

Comment

* getMessage()

**Method:** getMessage()

**Description:** Gets the text content of the message

**PreConditions:** N/A

**PostConditions:** The message’s content is returned to the control object

**Signature:** String getMessage()

* setMessage()

**Method:** setMessage()

**Description:** Sets the text content of the message

**PreConditions:** The message must meet guidelines

**PostConditions:** Content of message is changed

**Signature:** void setMessage( String text)

* getPoints()

**Method:** getPoints()

**Description:** Get the integer of the student’s reputation points.

**PreConditions:** N/A

**PostConditions:** An integer of points is returned and displayed.

**Signature:** Int getPoints()

* setPoints()

**Method:** setPoints()

**Description:** Set the student’s reputation points.

**PreConditions:** Point value can only be 1 or -1

**PostConditions:** Points are set.

**Signature:** void setPoints(int points)

Notification

* getNotification()

**Method:** getNotification()

**Description:** Returns the notification’s content as a String

**PreConditions:** N/A

**PostConditions:** The notification’s content is returned to the control object.

**Signature:** String getNotification()

* setNotification()

**Method:** setNotification()

**Description:** Edit a notification w/ string being passed in and save changes.

**PreConditions:** Notification has not been edited.

**PostConditions:** Notification has been edited.

**Signature:** void setNotification(String words)

* getTimestamp()

**Method:** getTimestamp()

**Description:** Returns the timestamp of a Notification object as a String

**PreConditions:** N/A

**PostConditions:** The timestamp of the notification is returned to the control object

**Signature:** String getTimestamp()

* setTimestamp()

**Method:** setTimestamp()

**Description:** Sets the timestamp of the Notification object

**PreConditions:** The timestamp being set to is in the correct format

**PostConditions:** Timestamp has been updated

**Signature:** Void setTimestamp(String timestamp)