Notification Class

```
Create a String variable called message
Create a int variable called date
Create a int variable called time
Create a User variable called receiver

getMessage()
    Return message

setMessage(message: String)
    Set message to the provided value

getReceiver()
    Return receiver

setReceiver(receiver: User)
    Set receiver to the provided value

getDateTime()
    Return date and time as one string variable

setDateTime(date: int, time:int)
    Set date and message to the provided values
```

Friend Class

```
Create a String called username
Create a Boolean called status
Create a list of content objects called watchedContent

getUsername()
    Return username

getStatus()
    Return status

getWatchedContent()
    Return watchedContent
```

User Class

```
Create a String called username
Create a String called email
Create a String called password
Create a Boolean called visible
Create a list of rating objects called rating
Create a list of search objects called searches
Create a list of friend objects called friends

getUsername()
    Return username

createAccount(email, username, password)
    Set email variable to email
    Set username variable to username
    Set password variable to password
    Return the User object

changeAccountDetails(username, email, password)
    If username input is not blank
    Then set username variable to username input

    If email input is not blank
    Then set email variable to email input

    If password input is not blank
    Then set password variable to password input

deleteFriend(friend: friend)
    Loop through friend list
    If friend exists, remove friend from list
    If not, then return an error message

getFriends()
    Return friend list

setVisible(Boolean)
    If true, visible equals true
    If false, visible equals false
```

```
clearHistory()
    Create a new list of search objects
    Set new list of search object equal to searches

login(username: String, password: String)

    if username input is equal to username variable and if password input is equal to password
    Then allow user to proceed

logOut()
    Exit

getSearches()
    Returns searches

addSearch(search: search)
    Adds the search object to the searches list

removeSearch(search: search)
    Loops through the searches list
    If search is there, remove search
    If not, do nothing
```

Search Class

Create a list of Keyword objects called keywords

```
getKeyword()
    Return keywords

setKeyword(keywords: Keyword)
    Adds the keyword object to the end of the keywords list

search(): result
    Create a temporary list of keywords named result
    Loops through the keywords list
    If item in keywords matches a keyword in a content object
    Add keyword to result
    Return result
```

Keyword Class

```
Create a String variable called value

getValue()
    Return value

setValue(value: String)
    Set value variable to value input
```

Content Class

```
Create a String variable called title
Create a String variable called description
Create a String variable called category
Create an int variable called date
Create a String list called streamingPlatforms
Create a list of keyword objects called contentKeyword
Create a list of Rating objects called contentRating
```

Constructor for **Content Class**

```
Content(title: String, description: String, category: String, date: int)
    Set title variable to title input
    Set description variable to description input
    Set category variable to category input
    Set date variable to date input

getTitle()
    Returns title

getCategory()
    Returns category

getDate()
    Returns date

getKeywords()
```

```
    Returns contentKeyword
addKeyword(keyword: keyword)
    Adds an imputed keyword object to the end of the contentKeyword list

removeKeyword(keyword:keyword)
    Loop through the contentKeyword list
    If inputted keyword matches keyword object in contentKeyword
    Remove inputted keyword

addStreamingPlatform(name: String)
    Add inputted name to the end of the streamingPlatform list

removeStreamingPlatform(name: String)
    Loop through the streamingPlatform list
    If inputted name matches a string in streamingPlatform list
    Remove inputted name

getRatingsForContent(content: content)
    Return contentRatings list

getAverageRatingsForContent(content: content)
    Create an int variable called temp
    Create an int variable called index
    Loop through contentRatings list
    For each iteration, add one to index
    For each object in the contentRatings list, use the getScore method and add the
    Integer value to temp
    Return the result of the division of temp by index-1
```

Rating Class

```
Create a float variable called score
Create a User object called User
Create a Content object called Content
```

Constructor for **Rating Class**

```
Rating(score: float, user: User, content: Content)
    Set score variable equal to score input
    Set user variable equal to User input
```

```
        Set content variable equal to Content input

getScore()
        Return score

setScore(score: float)
        Set score variable equal to score input

getUser()
        Return User

setContent(content: Content)
        Set content variable to Content input

getContent()
        Return content
```

ForYou Class

```
Create a list of Content objects called trendingContent
Create a String variable called preference

addContent(content: Content)
        Add content to the end of forYouContent list

removeContent(content:Content)
        Loop through the forYouContent list
        If there is a Content object that matches the inputted content
        Remove inputted content from forYouContent list

clearContent()
        Create a new list of Content objects
        Set new list of Content objects equal to forYouContent

setPreference(preference: String)
        Set the preference variable equal to the inputted preference
```

TrendingList Class

```
Create a new list of Content objects called trendingContent

addContent(content: Content)
    Loop through the trendingContent list
    At the end of the list, add the inputted Content

removeContent(content:Content)
    Loop through the trendingContent list
    If there is a Content object that matches the inputted content
    Remove inputted content from trendingContent list

sortByTrendiness()
    Create a temporary Content list known as tempList
    Create a temporary float known as tempFloat
    Loops through the trendingContent list
    For each Content object in the trendingContent list, use the getAverageRatingsForContent() method and record the result in tempFloat
    The highest scoring content object in trendingList is added to the beginning of tempList
    That same content object is removed from the trendingList

getTopContent()
    Loop through the trendingContent list
    Utilize the getAverageRatingsForContent() method for each object in the trendingContent list
    Return the Content object with the highest value

getTrendingContentByCategory(category: String)
    Create a temporary Content list known as tempList
    Loop through the trendingContent list
    Utilize the getCategory method on each object in the trendingContent list
    If the return value matches the inputted category, add the object to tempList
    Return tempList
```