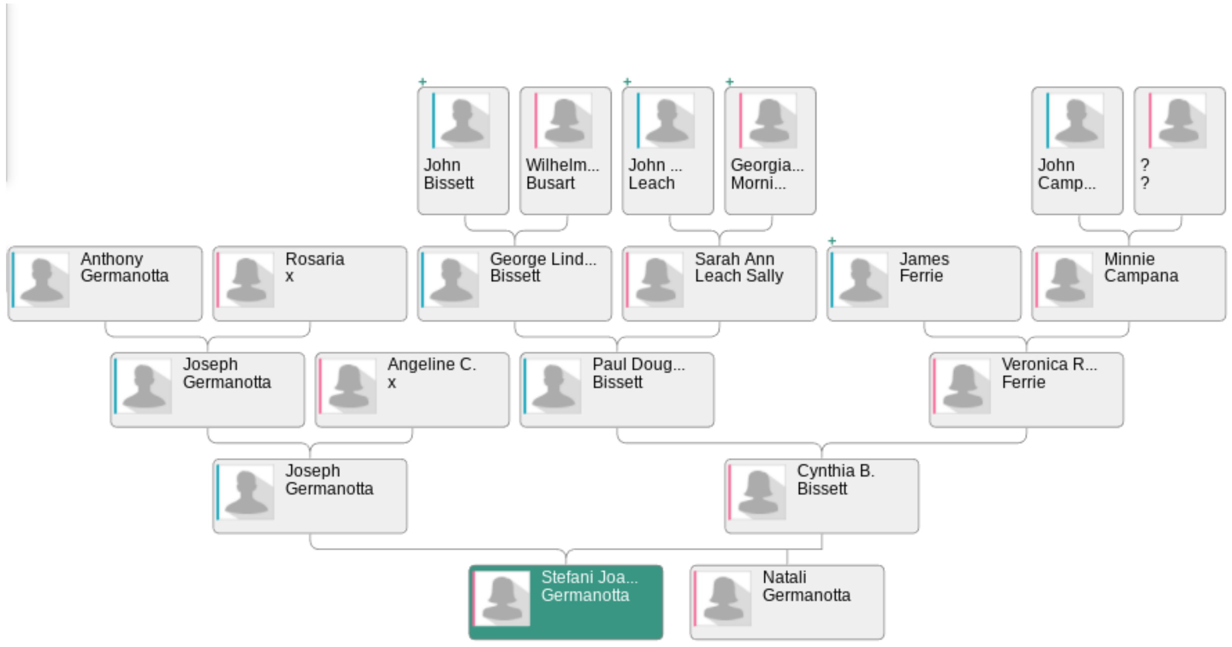
CSE1322L Assignment 8

Description

In this assignment you'll write a program to allow people to keep track of their family tree. As an example we'll be looking at Lady Gaga's family tree. According to geneanet.org her tree looks like this (Lady Gaga's real name is Stefani Joanne Angelina Germanotta):



This program will use a List (C#) or an ArrayList (Java) of Person, where a person has an ID, a Name and 2 parent IDs to hold their parents. When you add a new person to the tree, they will be given a unique number, and you'll be asked to select the number of each of their parents.

Assignment:

- Start by joining the following replit: https://replit.com/teams/join/glwzupcpssjxogrmyizarctzidafnwce-CSE1322LAssignment08
- You should see "CSE1322L_Assignment08_Java" and "CSE1322L_Assignment08_C#". Click the appropriate one to make your own copy of the base file.
  - You can do the assignment in replit, or in your favorite IDE. If you decide to do it in your favorite IDE, simply copy the 3 files (Family, Person and Main) from replit.
  - If you decide to do it in replit, remember you must still submit .java or .cs files. Do not submit a URL. Nor can you submit in replit.
- Familiarize yourself with the Person class contained in either Person.java or Person.cs.

Notice that each Person will have:
  - ID - sequentially assigned
  - Name - String
  - Parent1_id - int
  - Parent2_id - int
  - A constructor which sets the ID, name and parent IDs of this object
  - Getters for id, name, parent1 and parent2
  - A toString/ToString override that simply returns the object's name.
  - Note: If someone's parent is unknown you'd set it to -1.
- Next familiarize yourself with the Family class in Family.java or Family.cs
  - A family has an ArrayList (Java) or List (C#) of Person
  - You are provided with finished methods as follows:
    - addPerson which takes in a name and 2 parent IDs. It adds the person to the List/ArrayList
    - getIDFromName, which takes in a name and returns the ID of that person.
      - This is done with a loop over the ArrayList/List.
    - getNameFromID, which does the reverse of the last method, takes in a ID and returns the name of that person.
      - Again, done with a loop
    - parentRelationship, which takes in a String, and returns what that person's Parent would be called.
      - E.g. if passed "Self" it returns "Parent". If passed "Parent" it returns "Grand Parent", if passed "Great Great GrandParent" It returns "Great Great Great GrandParent"
    - childRelationship, which takes in a String, and returns what that person's child would be called.
      - E.g. If passed "Child" it'll return "GrandChild".
    - Everyone, which returns a single string with everyone's ID and name in it. This is used to produce menus so people can select their parents.
- Finally, familiarize yourself with the Main class (main.cs or Main.java). This method is complete and should not need to be changed in any way. This is a main method which adds a bunch of people to the tree (Lady Gaga's ancestors) and presents a menu that calls the appropriate functions in the Family objects.

<u>Your Coding Tasks:</u>

You do not need to change any provided code. It is all correct. You should only have to code the body of 4 methods in the Family.java or Family.cs file. The four methods you'll need to code are:

- getChildrenIDs which takes in an integer. This method should return direct children. For example in the picture of Lady Gaga's family above, if John Bissett was entered first, and has ID 0, if you called this method and passed it 0, it would return an ArrayList/List which contains the ID for George Lindsey Bissett as that is the only child of John Bissett.

    - This method can be written using a loop.
- getParents which takes in the ID of a child, and returns an Array with 2 cells, containing the ID of each of that child's parents. Note it may return -1 if we don't know the child's parents.
    - This method can be written using a loop
- printParents which takes in a childID and a relationship.
    - This method must **RECURSIVELY** print out all the parents, grand parents, great grandparents etc of the provided child.
    - Remember all recursive methods need a base condition. For this the base condition is that you are passed -1 as a childID you should do nothing since -1 implies unknown.
    - In all other conditions, you'll need to print out your parents, and their parents and their parents until you get to an unknown.
- printChildren which takes in a parentID and a relationship.
    - This method must RECURSIVELY print out all the children, grandchildren, great grandchildren etc of the provided parent.
    - Again, if you are passed -1, there is nothing to do.
    - Otherwise, you'll find all the direct children of this parent, and recurse to each child to find their descendants.

Once you have coded all 4 methods, you should be able to run it, and you'll see output similar to the sample output below

Sample Output:

```
What would you like to do
1. Add a person to the tree
2. See a person's ancestors
3. See a person's descendants
4. Quit
2
Who do you want to see ancestors of?
0) John Bissett
1) Wilhelm Busart
2) John Leach
3) Georgianna Morningstar
4) John Campana
5) Anthony Germanotta
6) Rosaria X
7) George Lindsey Bissett
8) Sarah Ann Leach Sally
9) James Ferrie
10) Minnie Campana
```

```
11) Joseph Germanotta
12) Angeline C
13) Paul Douglas Bissett
14) Veronica R Ferrie
15) Joseph A Germanotta
16) Cynthia B. Bissett
17) Stephanie Joanne Germanotta
18) Natali Germanotta

17
Self Stephanie Joanne Germanotta
Parent Joseph A Germanotta
GrandParent Joseph Germanotta
Great GrandParent Anthony Germanotta
Great GrandParent Rosaria X
GrandParent Angeline C
Parent Cynthia B. Bissett
GrandParent Paul Douglas Bissett
Great GrandParent George Lindsey Bissett
Great Great GrandParent John Bissett
Great Great GrandParent Wilhelm Busart
Great GrandParent Sarah Ann Leach Sally
Great Great GrandParent John Leach
Great Great GrandParent Georgianna Morningstar
GrandParent Veronica R Ferrie
Great GrandParent James Ferrie
Great GrandParent Minnie Campana
Great Great GrandParent John Campana
What would you like to do
1. Add a person to the tree
2. See a person's ancestors
3. See a person's descendants
4. Quit
3
Who do you want to see descendants of?
0) John Bissett
1) Wilhelm Busart
2) John Leach
3) Georgianna Morningstar
4) John Campana
5) Anthony Germanotta
6) Rosaria X
7) George Lindsey Bissett
8) Sarah Ann Leach Sally
9) James Ferrie
10) Minnie Campana
```

```
11) Joseph Germanotta
12) Angeline C
13) Paul Douglas Bissett
14) Veronica R Ferrie
15) Joseph A Germanotta
16) Cynthia B. Bissett
17) Stephanie Joanne Germanotta
18) Natali Germanotta

17
Self Stephanie Joanne Germanotta
What would you like to do
1. Add a person to the tree
2. See a person's ancestors
3. See a person's descendants
4. Quit
3
Who do you want to see descendants of?
0) John Bissett
1) Wilhelm Busart
2) John Leach
3) Georgianna Morningstar
4) John Campana
5) Anthony Germanotta
6) Rosaria X
7) George Lindsey Bissett
8) Sarah Ann Leach Sally
9) James Ferrie
10) Minnie Campana
11) Joseph Germanotta
12) Angeline C
13) Paul Douglas Bissett
14) Veronica R Ferrie
15) Joseph A Germanotta
16) Cynthia B. Bissett
17) Stephanie Joanne Germanotta
18) Natali Germanotta

0
Self John Bissett
Child George Lindsey Bissett
GrandChild Paul Douglas Bissett
Great GrandChild Cynthia B. Bissett
Great Great GrandChild Stephanie Joanne Germanotta
Great Great GrandChild Natali Germanotta
What would you like to do
```

```
1. Add a person to the tree
2. See a person's ancestors
3. See a person's descendants
4. Quit
2
Who do you want to see ancestors of?
0) John Bissett
1) Wilhelm Busart
2) John Leach
3) Georgianna Morningstar
4) John Campana
5) Anthony Germanotta
6) Rosaria X
7) George Lindsey Bissett
8) Sarah Ann Leach Sally
9) James Ferrie
10) Minnie Campana
11) Joseph Germanotta
12) Angeline C
13) Paul Douglas Bissett
14) Veronica R Ferrie
15) Joseph A Germanotta
16) Cynthia B. Bissett
17) Stephanie Joanne Germanotta
18) Natali Germanotta

0
Self John Bissett
What would you like to do
1. Add a person to the tree
2. See a person's ancestors
3. See a person's descendants
4. Quit
1
What is this person's name?
Baby Gaga
Who is their first parent
-1) Unknown
0) John Bissett
1) Wilhelm Busart
2) John Leach
3) Georgianna Morningstar
4) John Campana
5) Anthony Germanotta
6) Rosaria X
7) George Lindsey Bissett
```

```
8) Sarah Ann Leach Sally
9) James Ferrie
10) Minnie Campana
11) Joseph Germanotta
12) Angeline C
13) Paul Douglas Bissett
14) Veronica R Ferrie
15) Joseph A Germanotta
16) Cynthia B. Bissett
17) Stephanie Joanne Germanotta
18) Natali Germanotta

17
Who is their other parent
-1) Unknown
0) John Bissett
1) Wilhelm Busart
2) John Leach
3) Georgianna Morningstar
4) John Campana
5) Anthony Germanotta
6) Rosaria X
7) George Lindsey Bissett
8) Sarah Ann Leach Sally
9) James Ferrie
10) Minnie Campana
11) Joseph Germanotta
12) Angeline C
13) Paul Douglas Bissett
14) Veronica R Ferrie
15) Joseph A Germanotta
16) Cynthia B. Bissett
17) Stephanie Joanne Germanotta
18) Natali Germanotta

-1
What would you like to do
1. Add a person to the tree
2. See a person's ancestors
3. See a person's descendants
4. Quit
2
Who do you want to see ancestors of?
0) John Bissett
1) Wilhelm Busart
2) John Leach
```

```
3) Georgianna Morningstar
4) John Campana
5) Anthony Germanotta
6) Rosaria X
7) George Lindsey Bissett
8) Sarah Ann Leach Sally
9) James Ferrie
10) Minnie Campana
11) Joseph Germanotta
12) Angeline C
13) Paul Douglas Bissett
14) Veronica R Ferrie
15) Joseph A Germanotta
16) Cynthia B. Bissett
17) Stephanie Joanne Germanotta
18) Natali Germanotta
19) Baby Gaga

19
Self Baby Gaga
Parent Stephanie Joanne Germanotta
GrandParent Joseph A Germanotta
Great GrandParent Joseph Germanotta
Great Great GrandParent Anthony Germanotta
Great Great GrandParent Rosaria X
Great GrandParent Angeline C
GrandParent Cynthia B. Bissett
Great GrandParent Paul Douglas Bissett
Great Great GrandParent George Lindsey Bissett
Great Great Great GrandParent John Bissett
Great Great Great GrandParent Wilhelm Busart
Great Great GrandParent Sarah Ann Leach Sally
Great Great Great GrandParent John Leach
Great Great Great GrandParent Georgianna Morningstar
Great GrandParent Veronica R Ferrie
Great Great GrandParent James Ferrie
Great Great GrandParent Minnie Campana
Great Great Great GrandParent John Campana
What would you like to do
1. Add a person to the tree
2. See a person's ancestors
3. See a person's descendants
4. Quit
3
Who do you want to see descendants of?
0) John Bissett
```

```
1) Wilhelm Busart
2) John Leach
3) Georgianna Morningstar
4) John Campana
5) Anthony Germanotta
6) Rosaria X
7) George Lindsey Bissett
8) Sarah Ann Leach Sally
9) James Ferrie
10) Minnie Campana
11) Joseph Germanotta
12) Angeline C
13) Paul Douglas Bissett
14) Veronica R Ferrie
15) Joseph A Germanotta
16) Cynthia B. Bissett
17) Stephanie Joanne Germanotta
18) Natali Germanotta
19) Baby Gaga

19
Self Baby Gaga
What would you like to do
1. Add a person to the tree
2. See a person's ancestors
3. See a person's descendants
4. Quit
4
```

Submission Guidelines:

Program code (java or cs files only)

Please follow the posted submission guidelines here:
https://ccse.kennesaw.edu/fye/submissionguidelines.php

Ensure you submit before the deadline listed on the lab schedule for CSE1322L here:
https://ccse.kennesaw.edu/fye/courseschedules.php

Rubric:

Copied correctly (10 points total):

- Did not change any of the provided files (Person.java/Person.cs, Main.java/main.cs)

- Did not change any of the provided methods in Family.java/Family.cs except for the 4 methods below.

getChildrenIds (20 points total):
- Returns a List or ArrayList of ints (5 points)
- Successfully finds all direct children of the passed Parent ID (10 points)
- Does not contain duplicates (5 points)

getParents (20 points total):
- Returns an Array of 2 integers (10 points)
- Successfully finds both parents for a passed in Child ID (10 points)

printParents (25 points total):
- If not recursive? (-25 points)
- Correctly prints out the info about the passed person (5 points)
- Correct base condition that checks if we are passed -1 (5 points)
- Correctly figures out the relationship of each generation (5 points)
- Recurses on parent 1 (5 points)
- Recurses on parent 2 (5 points)

printChildren (25 points total):
- If recursion is not used to move between generations (-25 points)
- Correct base condition (5 points)
- Correctly prints out the info about the passed person (5 points)
- Correctly figures out the relationship for each generation (5 points)
- Gets all the children of the passed in person (5 points)
- Iterates over the children and recursively calls for each one (5 points)