# My Project

Generated by Doxygen 1.8.13

# Contents

# Chapter 1

# (IIT2015036) ppl-assignment

ppl-assignment-CosmicCoder96 ∗∗(Abhinav Khare)∗∗ created by GitHub Classroom

### About

>This project is an assignment for the Principal of Programming Languages course. It emphasises on Object Oriented Design and has been written in C++. Currently it holds the solution of *Questions 1 and 2.*

### Environment

>This project was built on MAC OS Sierra *Version 10.12.3 (16D32)* Text Editor used was sublime text 3 and command line tools were used. The documentation for the assignment can be found in Documentaion.pdf file and class diagram as under Class_Diagram.pdf. C++

#### g++ version

>4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.42.1)

### How to run?

#### Question 1

>First step would be to genereate a random list of boys and girls

```
g++ generate-girls.cpp -o data-girls
./data-girls
g++ generate-boys.cpp -o data-boys
./data-boys
```

>To run question 1

```
g++ 1.cpp Couple.cpp MakeCouples.cpp Boy.cpp Girl.cpp
./a.out
```

>The generated couples will be stored in List_of_Couples.txt

**Question 2**

>**Ignore if already done for question 1** : Generate random list of boys and girls

```
g++ generate-girls.cpp -o data-girls
./data-girls
g++ generate-boys.cpp -o data-boys
./data-boys
```

>Generate random gifts

```
g++ generate-gifts.cpp -o data-gifts
./data-gifts
```

>To run Querstion 2

```
g++ 2.cpp Couple.cpp MakeCouples.cpp Boy.cpp Girl.cpp
```

## Logging details

**General text files (Get recreated when the questions are run)**

- List of randomly generated boys : List_of_boys.txt
- List of randomly generated girls : List_of_Girls.txt
- List of randomly generated gifts : List_of_Gifts.cpp
- List of generated couples : List_of_Gifts.txt
- List of gifts given : List_of_Gifts_Given.txt >###Logging (Log file is permanent and maintains logs of all the executions) >It Maintains all the transactions which take place and logs them with timestamp.

**General Log** : Log.txt

# Chapter 2

# Class Index

## 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

# Chapter 3

# Class Documentation

## 3.1 Boy Class Reference

### Public Member Functions

- void getHappiness (Girl &girl_friend, int total_gifts)

  < *This is the class file for the Boy class. It defines it's various attributes of a boy object like attractiveness, budget, intelligence_level, and type which cam be MISER or GENEROUS OR GEEK*

### Public Attributes

- char **name** [MAX_NAME_LENGTH]
- int **attractiveness**
- int **budget**
- int **intelligence_level**
- int **min_attraction_requirement**
- int **type**
- int **committed**
- int **happiness**

The documentation for this class was generated from the following files:

- Boy.h
- Boy.cpp

## 3.2 Couple Class Reference

### Public Member Functions

- void find_compatibility ()

  < *This is the header file for the Couple class. It defines it's various attributes like happiness , compatible and various helper functions like finding kth most happy couple and kth most compatible couple*

### Static Public Member Functions

- static void k_happy (Couple *couples, int n, int k)
- static void k_compatible (Couple *couples, int n, int k)

  *The k_compatible function accepts an array of couples and the value entered by the user,k.*

### Public Attributes

- Boy **boy**
- Girl **girl**
- int **happiness**
- int **compatibility**

## 3.2.1 Member Function Documentation

#### 3.2.1.1 find_compatibility()

```
void Couple::find_compatibility ( )
```

< This is the header file for the Couple class. It defines it's various attributes like happiness , compatible and various helper functions like finding kth most happy couple and kth most compatible couple

The find_compatibility() function calculates the comaptibility of a particular couple by using the formula as given.

#### 3.2.1.2 k_compatible()

```
void Couple::k_compatible (
            Couple * couples,
            int n,
            int k )  [static]
```

The k_compatible function accepts an array of couples and the value entered by the user,k.

The function finds out the k most compatible couples and then prints them in order.

#### 3.2.1.3 k_happy()

```
void Couple::k_happy (
            Couple * couples,
            int n,
            int k )  [static]
```

The k_happy function accepts an array of couples and the value entered by the user,k

The documentation for this class was generated from the following files:

- Couple.h
- Couple.cpp

## 3.3 Gift Class Reference

### Public Attributes

- int **price**
- int **value**
- int **type**
- int **luxury_rating**
- int **difficulty**
- int **utility_value**
- int **utility_class**
- int **used**

The documentation for this class was generated from the following file:

- Gift.h

## 3.4 Girl Class Reference

### Public Member Functions

- void getHappiness (int total_gifts, int total_gift_value)

  *< The function getHappiness in the Girl class is used to calculate the happiness of a girl with total gift cost and total gift_value as parameters.*

### Public Attributes

- char **name** [MAX_NAME_LENGTH]
- int **attractiveness**
- int **maintainance_budget**
- int **intelligence_level**
- int **criterion**
- int **type**
- int **committed**
- int **happiness**

#### 3.4.1 Member Function Documentation

#### 3.4.1.1 getHappiness()

```
void Girl::getHappiness (
            int total_gifts,
            int total_gift_value )
```

< The function getHappiness in the Girl class is used to calculate the happiness of a girl with total gift cost and total gift_value as parameters.

< These if else conditions ensure that happiness of the girls is calculated based on the type of girl i.e. CHOOSY, NORMAL or DESPERATE

The documentation for this class was generated from the following files:

- Girl.h
- Girl.cpp

## 3.5 MakeCouples Class Reference

### Public Member Functions

- void generateCouples ()

  < *This is the class primarily responsible for the functioning of the program and makes use of Boy, Girl, Couple and Gift classes to solve problems.*
- void giveGifts ()

### Friends

- class **Couple**

### 3.5.1 Member Function Documentation

#### 3.5.1.1 generateCouples()

```
void MakeCouples::generateCouples ( )
```

< This is the class primarily responsible for the functioning of the program and makes use of Boy, Girl, Couple and Gift classes to solve problems.

< generateCouples() function is responsible for generating various random couples and store them in a specified text file.This function facilitates the transaction of various gifts which the boys in the couple gift to their girlfriends. < The file List_of_Boys.txt contains the list of randomly generated boys which is read and stored in the array.

< The file List_of_Girls.txt contains the list of randomly generated girls which is read and stored in the array.

< The Log.txt file maintains the log of all the events and transactions that take place including forming and break up of couples and gift transactions. It is a static file and maintains an over all log.

#### 3.5.1.2 giveGifts()

```
void MakeCouples::giveGifts ( )
```

< Couple list file is opened to accumulate all the couples in an array.

< List of gifts file is opened to accumulate all the gifts in an array

< suitable gifts are given by boys to the girls satisfying the restraints set upon by the boy type and

< These helper functions help in calculating the k most happy and k most compatible couples

The documentation for this class was generated from the following files:

- MakeCouples.h
- MakeCouples.cpp