My Project

Generated by Doxygen 1.8.13

Contents

Chapter 1

Class Index

1.1 Class List

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

Boy

BOY CLASS Contains the attributes of all the BOYS . . . . ??

Comm

COMM CLASS Makes suitable committed pairs . . . . ??

Couples

COUPLES CLASS Contains the attributes of all the COUPLES . . . . ??

Gift

GIFT CLASS Contains the attributes of all the GIFTS . . . . ??

Girl

GIRL CLASS Contains the attributes of all the GIRLS . . . . ??

Happy

HAPPY CLASS Calculates the Happiness and Compatibilty of Couples . . . . ??

makeGift

MakeGift CLASS Creates gifts of different Types . . . . ??

People

BOY CLASS Creates a random list of Boys and Girls with all the information needed to form Couples . . . . ??

Chapter 2

Class Documentation

2.1 Boy Class Reference

BOY CLASS Contains the attributes of all the BOYS.

```
#include <Boy.h>
```

Public Attributes

- int name

  Name of the BOY.
- int attRate

  Attraction Rate of the BOY.
- int budget

  Budget of the BOY.
- int intLevel

  Intelligence of the BOY.
- int minAttReq

  Minimum Attraction Requirement of the BOY.
- bool commitment

  Commitment of the BOY.

2.1.1 Detailed Description

BOY CLASS Contains the attributes of all the BOYS.

Attributes like: Name, Attraction Rate, Budget, Intelligence Level, Minimum Attraction Requirement, and Commitment.

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

- Boy.h

2.2 Comm Class Reference

COMM CLASS Makes suitable committed pairs.

```
#include <Comm.h>
```

Public Member Functions

- void pair (Boy b[ ], Girl g[ ], Couples r[ ], int ∗c, int noOfBoys, int noOfGirls, Couples P)

  Pairing up Girls with suitable Boys.

2.2.1 Detailed Description

COMM CLASS Makes suitable committed pairs.

Attributes like: Name, Attraction Rate, Maintenence Budget, Intelligence Level, Preference, and Commitment.

2.2.2 Member Function Documentation

2.2.2.1 pair()

```
void Comm::pair (
            Boy b[ ],
            Girl g[ ],
            Couples r[ ],
            int ∗ c,
            int noOfBoys,
            int noOfGirls,
            Couples P )
```

Pairing up Girls with suitable Boys.

Parameters

P	Pairing up Girls with suitable Boys

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

- Comm.h
- Comm.cpp

2.3 Couples Class Reference

COUPLES CLASS Contains the attributes of all the COUPLES.

```
#include <Couples.h>
```

Public Member Functions

- void commit (Boy p, Girl q)

  Committing a Boy and a Girl.
- void print (Couples r[ ], int c)

  Printing couples formed.

Public Attributes

- Boy **b**
- Girl **g**
- int bType

  1 - choosy, 2 - normal, 3 - desperate
- int gType

  1 - miser, 2 - generous, 3 - geek
- double chappy

  Happiness of Couple.
- double comp

  Compatibility of Couple.

2.3.1 Detailed Description

COUPLES CLASS Contains the attributes of all the COUPLES.

Attributes like: Girl Type, Boy Type, Happiness of Couple, and Compatibility of Couple.

2.3.2 Member Function Documentation

2.3.2.1 print()

```
void Couples::print (
            Couples r[ ],
            int c )
```

Printing couples formed.

Parameters

c	Printing couples formed.

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

- Couples.h
- Couples.cpp

2.4 Gift Class Reference

GIFT CLASS Contains the attributes of all the GIFTS.

```
#include <Gift.h>
```

Public Member Functions

- void makeBasket (Couples r[ ], int ∗c, Gift gt[ ], Happy hp)

  Making a gift basket according to the couple.

Public Attributes

- int item

  Gift Name.
- int price

  Gift Price.
- int value

  Gift Value.
- int luxury

  Luxury.
- int diffToGet

  Difficulty To Get.
- int utilityVal

  Utility Value.
- int utilityClass

  Utility Class.

2.4.1 Detailed Description

GIFT CLASS Contains the attributes of all the GIFTS.

Attributes like: Gift Name, Price, Value, Luxury, Difficulty To Get, Utility Value, and Utility Class.

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

- Gift.h
- Gift.cpp

2.5 Girl Class Reference

GIRL CLASS Contains the attributes of all the GIRLS.

```
#include <Girl.h>
```

Public Attributes

- int name

  Name of the GIRL.
- int attRate

  Attraction Rate of the GIRL.
- int maintBudget

  MAintenence Budget of the GIRL.
- int intLevel

  Intelligence of the GIRL.
- int preference

  Preference of the GIRL : 1 - MOST ATTRACTIVE, 2 - RICHEST, 3 - MOST INTELLIGENT.
- bool commitment

  Commitment of the GIRL.

2.5.1 Detailed Description

GIRL CLASS Contains the attributes of all the GIRLS.

Attributes like: Name, Attraction Rate, Maintenence Budget, Intelligence Level, Preference, and Commitment.

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

- Girl.h

2.6 Happy Class Reference

HAPPY CLASS Calculates the Happiness and Compatibilty of Couples.

```
#include <Happy.h>
```

Public Member Functions

- void Happiness (Couples r[ ], int c)

  Calculates the Happiness and Compatibility of Couples.

2.6.1 Detailed Description

HAPPY CLASS Calculates the Happiness and Compatibilty of Couples.

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

- Happy.h
- Happy.cpp

2.7 makeGift Class Reference

makeGift CLASS Creates gifts of different Types.

```
#include <makeGift.h>
```

Public Member Functions

- void makeGifts ()

  Randomly generates various types of gifts.

2.7.1 Detailed Description

makeGift CLASS Creates gifts of different Types.

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

- makeGift.h
- makeGift.cpp

2.8 People Class Reference

BOY CLASS Creates a random list of Boys and Girls with all the information needed to form Couples.

```
#include <People.h>
```

Public Member Functions

- void getPeople ()

  Generates a random list of Boys and Girls with all the information needed to form Couples.

2.8.1 Detailed Description

BOY CLASS Creates a random list of Boys and Girls with all the information needed to form Couples.

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

- People.h
- People.cpp