# PPL Assignment

IIT2015099

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:

# Chapter 2

# File Index

## 2.1 File List

Here is a list of all files with brief descriptions:

# Chapter 3

# Class Documentation

## 3.1 boys Class Reference

### Public Member Functions

- int readboyscount ()
- int input (boys *boyss, int nb)

  *boys data input.*
- int logging (boys *boyss, int nb)

  *inserts girlfriend for a boyfriend if exists into log file.*

### Public Attributes

- std::string name
- std::string type
- std::string girlname
- int attractiveness
- int intelligence
- int budget
- int happiness
- int committed
- int min_attractive

### 3.1.1 Detailed Description

Definition at line 2 of file boys.cpp.

### 3.1.2 Member Function Documentation

#### 3.1.2.1 input()

```
int boys::input (
            boys * boyss,
            int nb )  [inline]
```

boys data input.

Definition at line 18 of file boys.cpp.

#### 3.1.2.2 logging()

```
int boys::logging (
            boys * boyss,
            int nb )  [inline]
```

inserts girlfriend for a boyfriend if exists into log file.

Definition at line 32 of file boys.cpp.

#### 3.1.2.3 readboyscount()

```
int boys::readboyscount ( )  [inline]
```

Increment count if this character is newline.

number of couples.

Definition at line 7 of file boys.cpp.

### 3.1.3 Member Data Documentation

#### 3.1.3.1 attractiveness

```
int boys::attractiveness
```

Definition at line 6 of file boys.cpp.

#### 3.1.3.2 budget

```
int boys::budget
```

Definition at line 6 of file boys.cpp.

#### 3.1.3.3 committed

```
int boys::committed
```

Definition at line 6 of file boys.cpp.

#### 3.1.3.4 girlname

```
std::string boys::girlname
```

Definition at line 5 of file boys.cpp.

#### 3.1.3.5 happiness

```
int boys::happiness
```

Definition at line 6 of file boys.cpp.

#### 3.1.3.6 intelligence

```
int boys::intelligence
```

Definition at line 6 of file boys.cpp.

#### 3.1.3.7 min_attractive

```
int boys::min_attractive
```

Definition at line 6 of file boys.cpp.

#### 3.1.3.8 name

```
std::string boys::name
```

Definition at line 5 of file boys.cpp.

#### 3.1.3.9 type

```
std::string boys::type
```

Definition at line 5 of file boys.cpp.

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

- PPL/ques2/boys.cpp

## 3.2 couples Class Reference

### Public Member Functions

- int input (couples *couple, int count)
- int input1 (couples *couple, int count)

  *data read.*
- int readcouplecount ()
- int pairing (boys *boyss, girls *girlss, int nb, int ng)

  *Pairing.*
- int couplegifting (couples *couple, int count, gifts *gif, int ngf)

  *Gift Exchanges.*
- int mosthappy (couples *couple, int count, int k)

  *bubble sort for happiness.*
- int mostcompatible (couples *couple, int count, int k)

  *bubble sort for compatibility.*

### Public Attributes

- std::string bname
- std::string btype
- std::string gname
- std::string gtype
- int bbud
- int gbud
- int batt
- int gatt
- int bint
- int gint
- int compatibility
- double happiness

### 3.2.1 Detailed Description

Definition at line 8 of file couples.cpp.

### 3.2.2 Member Function Documentation

#### 3.2.2.1 couplegifting()

```
int couples::couplegifting (
            couples * couple,
            int count,
            gifts * gif,
            int ngf )  [inline]
```

Gift Exchanges.

Definition at line 127 of file couples.cpp.

#### 3.2.2.2 input()

```
int couples::input (
            couples * couple,
            int count )  [inline]
```

data read of couples.

Definition at line 14 of file couples.cpp.

#### 3.2.2.3 input1()

```
int couples::input1 (
            couples * couple,
            int count )  [inline]
```

data read.

Definition at line 29 of file couples.cpp.

#### 3.2.2.4 mostcompatible()

```
int couples::mostcompatible (
            couples * couple,
            int count,
            int k )  [inline]
```

bubble sort for compatibility.

Definition at line 251 of file couples.cpp.

#### 3.2.2.5 mosthappy()

```
int couples::mosthappy (
            couples * couple,
            int count,
            int k )  [inline]
```

bubble sort for happiness.

Definition at line 231 of file couples.cpp.

#### 3.2.2.6 pairing()

```
int couples::pairing (
            boys * boyss,
            girls * girlss,
            int nb,
            int ng )  [inline]
```

Pairing.

Definition at line 54 of file couples.cpp.

#### 3.2.2.7 readcouplecount()

```
int couples::readcouplecount ( )  [inline]
```

Increment count if this character is newline.

number of couples.

Definition at line 43 of file couples.cpp.

### 3.2.3 Member Data Documentation

#### 3.2.3.1 batt

```
int couples::batt
```

Definition at line 12 of file couples.cpp.

#### 3.2.3.2 bbud

```
int couples::bbud
```

Definition at line 12 of file couples.cpp.

#### 3.2.3.3 bint

```
int couples::bint
```

Definition at line 12 of file couples.cpp.

#### 3.2.3.4 bname

```
std::string couples::bname
```

Definition at line 11 of file couples.cpp.

#### 3.2.3.5 btype

```
std::string couples::btype
```

Definition at line 11 of file couples.cpp.

#### 3.2.3.6 compatibility

```
int couples::compatibility
```

Definition at line 12 of file couples.cpp.

#### 3.2.3.7 gatt

```
int couples::gatt
```

Definition at line 12 of file couples.cpp.

#### 3.2.3.8 gbud

```
int couples::gbud
```

Definition at line 12 of file couples.cpp.

#### 3.2.3.9 gint

```
int couples::gint
```

Definition at line 12 of file couples.cpp.

#### 3.2.3.10 gname

```
std::string couples::gname
```

Definition at line 11 of file couples.cpp.

#### 3.2.3.11 gtype

```
std::string couples::gtype
```

Definition at line 11 of file couples.cpp.

#### 3.2.3.12 happiness

```
double couples::happiness
```

Definition at line 13 of file couples.cpp.

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

- PPL/ques2/couples.cpp

## 3.3 gifts Class Reference

### Public Member Functions

- int readgiftscount ()
- int input (gifts *gif, int ngf)

  *reading gifts data.*

### Public Attributes

- std::string type
- int value

  *attributes of gifts.*
- int price

### 3.3.1 Detailed Description

Definition at line 1 of file gifts.cpp.

### 3.3.2 Member Function Documentation

#### 3.3.2.1 input()

```
int gifts::input (
            gifts * gif,
            int ngf )  [inline]
```

reading gifts data.

Definition at line 17 of file gifts.cpp.

#### 3.3.2.2 readgiftscount()

```
int gifts::readgiftscount ( )  [inline]
```

Increment count if this character is newline.

number of couples.

Definition at line 6 of file gifts.cpp.

### 3.3.3 Member Data Documentation

#### 3.3.3.1 price

```
int gifts::price
```

Definition at line 5 of file gifts.cpp.

#### 3.3.3.2 type

```
std::string gifts::type
```

Definition at line 4 of file gifts.cpp.

#### 3.3.3.3 value

```
int gifts::value
```

attributes of gifts.

Definition at line 5 of file gifts.cpp.

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

- PPL/ques2/gifts.cpp

## 3.4 girls Class Reference

### Public Member Functions

- int readgirlscount ()
- int input (girls *girlss, int ng)

  *reading girls data.*

### Public Attributes

- std::string name
- std::string type
- std::string boyname
- std::string need
- int attractiveness

  *attributes of girls.*
- int maintenance
- int intelligence
- int happiness
- int committed

## 3.4.1 Detailed Description

Definition at line 1 of file girls.cpp.

## 3.4.2 Member Function Documentation

### 3.4.2.1 input()

```
int girls::input (
            girls * girlss,
            int ng )  [inline]
```

reading girls data.

Definition at line 17 of file girls.cpp.

### 3.4.2.2 readgirlscount()

```
int girls::readgirlscount ( )  [inline]
```

Increment count if this character is newline.

number of couples.

Definition at line 6 of file girls.cpp.

## 3.4.3 Member Data Documentation

#### 3.4.3.1 attractiveness

```
int girls::attractiveness
```

attributes of girls.

Definition at line 5 of file girls.cpp.

#### 3.4.3.2 boyname

```
std::string girls::boyname
```

Definition at line 4 of file girls.cpp.

#### 3.4.3.3 committed

```
int girls::committed
```

Definition at line 5 of file girls.cpp.

#### 3.4.3.4 happiness

```
int girls::happiness
```

Definition at line 5 of file girls.cpp.

#### 3.4.3.5 intelligence

```
int girls::intelligence
```

Definition at line 5 of file girls.cpp.

#### 3.4.3.6 maintenance

```
int girls::maintenance
```

Definition at line 5 of file girls.cpp.

#### 3.4.3.7 name

```
std::string girls::name
```

Definition at line 4 of file girls.cpp.

#### 3.4.3.8 need

```
std::string girls::need
```

Definition at line 4 of file girls.cpp.

#### 3.4.3.9 type

```
std::string girls::type
```

Definition at line 4 of file girls.cpp.

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

- PPL/ques2/girls.cpp

## 3.5 util Class Reference

### Public Member Functions

- int coupling ()
- int gifting ()
- int most (int k)

### 3.5.1 Detailed Description

Definition at line 1 of file util.cpp.

### 3.5.2 Member Function Documentation

#### 3.5.2.1 coupling()

```
int util::coupling ( )  [inline]
```

taking boys input from boys.txt .

taking girls input from boys.txt.

pairing girl-boys if attractive of girl is greater than boy's reqquirement, satisfying the budget of boy and boys fall under the selection criterion of girl.

inserting into log file relations of a boy.

Definition at line 4 of file util.cpp.

#### 3.5.2.2 gifting()

```
int util::gifting ( )  [inline]
```

counting the number of couples.

Reading couples data from couple.txt.

Reading the types of gifts.

Gift exchanges,happiness and compatibility calculation and inserting into log file and fcalc.txt.

Definition at line 19 of file util.cpp.

#### 3.5.2.3 most()

```
int util::most (
            int k )  [inline]
```

counting the number of couples.

Reading the happiness and compatibility of couples in couples∗ coup.

find the k-most happy couple.

find the k most compatiblee couple.

Definition at line 32 of file util.cpp.

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

- PPL/ques2/util.cpp

# Chapter 4

# File Documentation

## 4.1 PPL/ques2/boys.cpp File Reference

```
#include <fstream>
```

### Classes

- class boys

## 4.2 PPL/ques2/couples.cpp File Reference

```
#include "girls.cpp"
#include "boys.cpp"
#include "gifts.cpp"
#include <fstream>
#include <ctime>
#include <math.h>
```

### Classes

- class couples

## 4.3 PPL/ques2/gifts.cpp File Reference

### Classes

- class gifts

## 4.4 PPL/ques2/girls.cpp File Reference

### Classes

- class girls

## 4.5 PPL/ques2/ques2.cpp File Reference

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include "couples.cpp"
#include "util.cpp"
```

### Functions

- int main (int argc, char **argv)

### 4.5.1 Function Documentation

#### 4.5.1.1 main()

```
int main (
            int argc,
            char ** argv )
```

Inserting the couples formed into log file and couples.txt

Inserting happiness and compatibility into fcalc.txt

Printing the k happiest and k compatible couples.

Definition at line 7 of file ques2.cpp.

## 4.6 PPL/ques2/randomgen.cpp File Reference

```
#include <iostream>
#include "gifts.cpp"
```

### Functions

- int main (int argc, char **argv)

### 4.6.1 Function Documentation

#### 4.6.1.1 main()

```
int main (
            int argc,
            char ** argv )
```

Randomly Generating different types of boys in boys.txt.

boy name.

boy type.

attractiveness.

intelligent.

budget.

minimum attr.

Randomly Generating different types of girls in girls.txt.

Name.

type.

type.

attractiveness.

intelligent.

maintenance.

different types of gift int gift.txt.

type.

Price.

Value.

luxury gifts will have more Price.

Value.

Generating the gifts in an srted order of their price.

if gift is luxury keeping it in luxury.txt as well.

Definition at line 3 of file randomgen.cpp.

## 4.7 PPL/ques2/util.cpp File Reference

### Classes

- class util