

PPL Assignment - Question 5

IIT2015099

Generated by Doxygen 1.8.13

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	attributes Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Member Data Documentation	7
4.1.2.1	attractiveness	7
4.1.2.2	committed	8
4.1.2.3	happiness	8
4.1.2.4	intelligence	8
4.1.2.5	name	8
4.1.2.6	type	8
4.2	boys Class Reference	8
4.2.1	Detailed Description	9
4.2.2	Member Function Documentation	9
4.2.2.1	input()	9
4.2.2.2	logging()	9
4.2.2.3	readboyscount()	10

4.2.2.4	sorti()	10
4.2.3	Member Data Documentation	10
4.2.3.1	budget	10
4.2.3.2	girlName	10
4.2.3.3	min_attractive	10
4.3	couples Class Reference	11
4.3.1	Detailed Description	11
4.3.2	Member Function Documentation	11
4.3.2.1	couplegifting()	11
4.3.2.2	input()	12
4.3.2.3	input1()	12
4.3.2.4	mosthappy()	12
4.3.2.5	pairingq1()	12
4.3.2.6	pairingq5()	13
4.3.2.7	readcouplecount()	13
4.3.3	Member Data Documentation	13
4.3.3.1	batt	13
4.3.3.2	bbud	13
4.3.3.3	bint	14
4.3.3.4	bname	14
4.3.3.5	btype	14
4.3.3.6	compatibility	14
4.3.3.7	gatt	14
4.3.3.8	gbud	14
4.3.3.9	gint	15
4.3.3.10	gname	15
4.3.3.11	gtype	15
4.3.3.12	happiness	15
4.4	gifts Class Reference	15
4.4.1	Detailed Description	16

4.4.2	Member Function Documentation	16
4.4.2.1	input()	16
4.4.2.2	readgiftscount()	16
4.4.3	Member Data Documentation	16
4.4.3.1	price	16
4.4.3.2	type	16
4.4.3.3	value	17
4.5	girls Class Reference	17
4.5.1	Detailed Description	17
4.5.2	Member Function Documentation	17
4.5.2.1	input()	18
4.5.2.2	readgirlscount()	18
4.5.2.3	sorti()	18
4.5.3	Member Data Documentation	18
4.5.3.1	boyname	18
4.5.3.2	maintenance	18
4.5.3.3	need	19
4.6	util Class Reference	19
4.6.1	Detailed Description	19
4.6.2	Member Function Documentation	19
4.6.2.1	coupling()	19
4.6.2.2	gifting()	20
4.6.2.3	most()	20
5	File Documentation	21
5.1	PPL/ques5/attributes.cpp File Reference	21
5.2	PPL/ques5/boys.cpp File Reference	21
5.3	PPL/ques5/couples.cpp File Reference	21
5.4	PPL/ques5/gifts.cpp File Reference	22
5.5	PPL/ques5/girls.cpp File Reference	22
5.6	PPL/ques5/main.cpp File Reference	22
5.6.1	Function Documentation	22
5.6.1.1	main()	22
5.7	PPL/ques5/randomgen.cpp File Reference	23
5.7.1	Function Documentation	23
5.7.1.1	main()	23
5.8	PPL/ques5/util.cpp File Reference	24

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

attributes	7
boys	8
girls	17
couples	11
gifts	15
util	19

Chapter 2

Class Index

2.1 Class List

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

attributes	7
boys	8
couples	11
gifts	15
girls	17
util	19

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

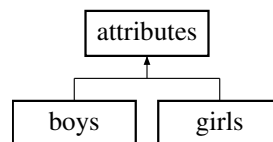
PPL/ques5/ attributes.cpp	21
PPL/ques5/ boys.cpp	21
PPL/ques5/ couples.cpp	21
PPL/ques5/ gifts.cpp	22
PPL/ques5/ girls.cpp	22
PPL/ques5/ main.cpp	22
PPL/ques5/ randomgen.cpp	23
PPL/ques5/ util.cpp	24

Chapter 4

Class Documentation

4.1 attributes Class Reference

Inheritance diagram for attributes:



Public Attributes

- std::string [name](#)
- std::string [type](#)
- int [attractiveness](#)
- int [intelligence](#)
- int [happiness](#)
- int [committed](#)

4.1.1 Detailed Description

Definition at line 1 of file attributes.cpp.

4.1.2 Member Data Documentation

4.1.2.1 attractiveness

```
int attributes::attractiveness
```

Definition at line 5 of file attributes.cpp.

4.1.2.2 committed

```
int attributes::committed
```

Definition at line 5 of file attributes.cpp.

4.1.2.3 happiness

```
int attributes::happiness
```

Definition at line 5 of file attributes.cpp.

4.1.2.4 intelligence

```
int attributes::intelligence
```

Definition at line 5 of file attributes.cpp.

4.1.2.5 name

```
std::string attributes::name
```

Definition at line 4 of file attributes.cpp.

4.1.2.6 type

```
std::string attributes::type
```

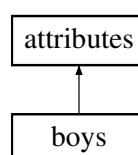
Definition at line 4 of file attributes.cpp.

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

- PPL/ques5/[attributes.cpp](#)

4.2 boys Class Reference

Inheritance diagram for boys:



Public Member Functions

- int [readboyscount](#) ()
- int [input](#) ([boys](#) *boyss, int nb)
boys data input.
- void [sorti](#) ([boys](#) *boyss, int nb)
- int [logging](#) ([boys](#) *boyss, int nb)
inserts girlfriend for a boyfriend if exists into log file.

Public Attributes

- std::string [girlname](#)
- int [budget](#)
- int [min_attractive](#)

4.2.1 Detailed Description

Definition at line 2 of file boys.cpp.

4.2.2 Member Function Documentation

4.2.2.1 [input\(\)](#)

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

boys data input.

Definition at line 18 of file boys.cpp.

4.2.2.2 [logging\(\)](#)

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

inserts girlfriend for a boyfriend if exists into log file.

Definition at line 40 of file boys.cpp.

4.2.2.3 readboyscount()

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

Increment count if this character is newline.

number of couples.

Definition at line 7 of file boys.cpp.

4.2.2.4 sorti()

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

Definition at line 32 of file boys.cpp.

4.2.3 Member Data Documentation

4.2.3.1 budget

```
int boys::budget
```

Definition at line 6 of file boys.cpp.

4.2.3.2 girlname

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

Definition at line 5 of file boys.cpp.

4.2.3.3 min_attractive

```
int boys::min_attractive
```

Definition at line 6 of file boys.cpp.

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

- [PPL/ques5/boys.cpp](#)

4.3 couples Class Reference

Public Member Functions

- int `input` (`couples` *couple, int count)
- int `input1` (`couples` *couple, int count)
data read.
- int `readcouplecount` ()
- int `pairingq1` (`boys` *boyss, `girls` *girlss, int nb, int ng)
Pairing.
- int `pairingq5` (`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.

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`

4.3.1 Detailed Description

Definition at line 9 of file couples.cpp.

4.3.2 Member Function Documentation

4.3.2.1 couplegifting()

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

Gift Exchanges.

Definition at line 233 of file couples.cpp.

4.3.2.2 input()

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

data read of couples.

Definition at line 15 of file couples.cpp.

4.3.2.3 input1()

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

data read.

Definition at line 30 of file couples.cpp.

4.3.2.4 mosthappy()

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

bubble sort for happiness.

Definition at line 337 of file couples.cpp.

4.3.2.5 pairingq1()

```
int couples::pairingq1 (
    boys * boyss,
    girls * girlss,
    int nb,
    int ng ) [inline]
```

Pairing.

Definition at line 55 of file couples.cpp.

4.3.2.6 pairingq5()

```
int couples::pairingq5 (
    boys * boyss,
    girls * girlss,
    int nb,
    int ng ) [inline]
```

Pairing.

First a girl is given a chance.

The boy is given a chance here.

Definition at line 128 of file couples.cpp.

4.3.2.7 readcouplecount()

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

Increment count if this character is newline.

number of couples.

Definition at line 44 of file couples.cpp.

4.3.3 Member Data Documentation

4.3.3.1 batt

```
int couples::batt
```

Definition at line 13 of file couples.cpp.

4.3.3.2 bbud

```
int couples::bbud
```

Definition at line 13 of file couples.cpp.

4.3.3.3 bint

```
int couples::bint
```

Definition at line 13 of file couples.cpp.

4.3.3.4 bname

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

Definition at line 12 of file couples.cpp.

4.3.3.5 btype

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

Definition at line 12 of file couples.cpp.

4.3.3.6 compatibility

```
int couples::compatibility
```

Definition at line 13 of file couples.cpp.

4.3.3.7 gatt

```
int couples::gatt
```

Definition at line 13 of file couples.cpp.

4.3.3.8 gbud

```
int couples::gbud
```

Definition at line 13 of file couples.cpp.

4.3.3.9 gint

```
int couples::gint
```

Definition at line 13 of file couples.cpp.

4.3.3.10 gname

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

Definition at line 12 of file couples.cpp.

4.3.3.11 gtype

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

Definition at line 12 of file couples.cpp.

4.3.3.12 happiness

```
double couples::happiness
```

Definition at line 14 of file couples.cpp.

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

- PPL/ques5/[couples.cpp](#)

4.4 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](#)

4.4.1 Detailed Description

Definition at line 1 of file gifts.cpp.

4.4.2 Member Function Documentation

4.4.2.1 input()

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

reading gifts data.

Definition at line 17 of file gifts.cpp.

4.4.2.2 readgiftscount()

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

Increment count if this character is newline.

number of couples.

Definition at line 6 of file gifts.cpp.

4.4.3 Member Data Documentation

4.4.3.1 price

```
int gifts::price
```

Definition at line 5 of file gifts.cpp.

4.4.3.2 type

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

Definition at line 4 of file gifts.cpp.

4.4.3.3 value

```
int gifts::value
```

attributes of gifts.

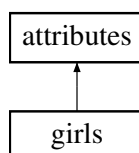
Definition at line 5 of file gifts.cpp.

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

- [PPL/ques5/gifts.cpp](#)

4.5 girls Class Reference

Inheritance diagram for girls:



Public Member Functions

- int [readgirlscount](#) ()
- void [sorti](#) ([girls](#) *girlss, int ng)
- int [input](#) ([girls](#) *girlss, int ng)

reading girls data.

Public Attributes

- std::string [boyname](#)
- std::string [need](#)
- int [maintenance](#)

attributes of girls.

4.5.1 Detailed Description

Definition at line 1 of file girls.cpp.

4.5.2 Member Function Documentation

4.5.2.1 input()

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

reading girls data.

Definition at line 27 of file girls.cpp.

4.5.2.2 readgirlscount()

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

Increment count if this character is newline.

number of couples.

Definition at line 6 of file girls.cpp.

4.5.2.3 sorti()

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

Definition at line 17 of file girls.cpp.

4.5.3 Member Data Documentation

4.5.3.1 boyname

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

Definition at line 4 of file girls.cpp.

4.5.3.2 maintenance

```
int girls::maintenance
```

attributes of girls.

Definition at line 5 of file girls.cpp.

4.5.3.3 need

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

Definition at line 4 of file girls.cpp.

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

- PPL/ques5/[girls.cpp](#)

4.6 util Class Reference

Public Member Functions

- int [coupling](#) ()
- int [gifting](#) ()
- int [most](#) (int k)

4.6.1 Detailed Description

Definition at line 1 of file util.cpp.

4.6.2 Member Function Documentation

4.6.2.1 coupling()

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

taking boys input from boys.txt .

taking girls input from boys.txt.

sorting boys in order of attractiveness.

sorting girls in order of maintenance.

Used a rand() function to determine which function to use.

pairing girl-boys if attractive of girl is greater than boy's requirement, 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.

4.6.2.2 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 31 of file util.cpp.

4.6.2.3 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.

Definition at line 44 of file util.cpp.

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

- [PPL/ques5/util.cpp](#)

Chapter 5

File Documentation

5.1 PPL/ques5/attributes.cpp File Reference

Classes

- class [attributes](#)

5.2 PPL/ques5/boys.cpp File Reference

```
#include <fstream>
```

Classes

- class [boys](#)

5.3 PPL/ques5/couples.cpp File Reference

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

Classes

- class [couples](#)

5.4 PPL/ques5/gifts.cpp File Reference

Classes

- class [gifts](#)

5.5 PPL/ques5/girls.cpp File Reference

Classes

- class [girls](#)

5.6 PPL/ques5/main.cpp File Reference

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

Functions

- int [main](#) (int argc, char **argv)

5.6.1 Function Documentation

5.6.1.1 main()

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

Inserting the couples formed into log file and couples.txt

Inserting happiness and compatibility into fcalc.txt

taking random value of k.

Printing the k happiest and k compatible couples.

Definition at line 7 of file main.cpp.

5.7 PPL/ques5/randomgen.cpp File Reference

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

Functions

- int [main](#) (int argc, char **argv)

5.7.1 Function Documentation

5.7.1.1 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 sorted order of their price.

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

Definition at line 3 of file randomgen.cpp.

5.8 PPL/ques5/util.cpp File Reference

Classes

- class [util](#)