

## My Project

Generated by Doxygen 1.8.13



# Contents



# Chapter 1

## ppl-assignment-ritikamotwani

ppl-assignment-ritikamotwani created by GitHub Classroom

**Ritika Motwani - IIT2015096**

### ###Language Used

C++

The project was made using Linux Ubuntu and the documentation was made through doxygen and latex with the class diagram converted to pdf format

### ### Files

.h files are the header files made and .cpp contain functions  
.txt files contain the output,the randomly generated list of the input parameters  
.pdf is for the documentation and Class diagram

### For Execution

To execute Question1 and Question2:

```
g++ -c girl.cpp
g++ -c boy.cpp
g++ -c couple.cpp
g++ -c make_pair.cpp
g++ -c boy_list.cpp
g++ -c girl_list.cpp
g++ -c gift_list.cpp
ar rvs ritika_file.a boy.o couple.o girl.o boy_list.o gift_list.o girl_list.o make_pair.o
```

Then

```
g++ ques1.cpp ritika_file.a
./a.out
g++ ques2_main.cpp ritika_file.a
./a.out
and input the value of k
Output will be there in couple.txt and happ_comp_gift.
```

For all the questions from three to ten

The List of couples , randomly generated girls, boys and gifts should exist That Is  
#### Question1 should be complied once before following the execution steps of any of the below questions.

Question3 :

```
For compilation- g++ ques3.cpp inh_pair.cpp couple.cpp boy.cpp girl.cpp Fill.cpp
For execution- ./a.out
Give the value of k as input.
```

Output- happ\_comp.txt

Question4 :

For compilation- g++ ques4.cpp inh\_pair.cpp couple.cpp boy.cpp girl.cpp Fill.cpp  
For execution- ./a.out  
Give the value of k as input.  
Output- new\_couple.txt

Question5 :

For compilation- g++ ques5.cpp girl.cpp boy.cpp couple.cpp new\_order\_couple.h  
Execution- ./a.out  
Output- order\_couple.txt  
For k happy couples compile and execute question2.

Question6 :

For compilation- g++ ques6.cpp inh\_pair.cpp couple.cpp boy.cpp girl.cpp Fill.cpp t\_days.cpp  
Execution- ./a.out  
Output- ques6.txt

Question7 :

For compilation- g++ ques7.cpp inh\_pair.cpp couple.cpp boy.cpp girl.cpp Fill.cpp Sorted\_Class.cpp  
hash.cpp  
Execution- ./a.out  
Output- Three\_Ways.txt

Question8 :

For compilation- g++ ques8.cpp make\_pair.cpp NewWay.cpp girl.cpp boy.cpp couple.cpp  
For execution- ./a.out  
Output- happ\_comp\_gift

Question9 :

For Compilation- g++ ques9.cpp secondary.cpp girl.cpp boy.cpp  
Execution- ./a.out  
Output- ques9.txt and ques9\_gifts.txt

## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

Array_Class . . . . .	??
Boy . . . . .	??
Geek . . . . .	??
Generous . . . . .	??
Miser . . . . .	??
Couple . . . . .	??
Fill . . . . .	??
Gift . . . . .	??
Girl . . . . .	??
Choosy . . . . .	??
Desperate . . . . .	??
Normal . . . . .	??
hash . . . . .	??
MakePair . . . . .	??
NewCouple . . . . .	??
NewWay . . . . .	??
Pair . . . . .	??
secondary . . . . .	??
Sorted_Class . . . . .	??
t_days . . . . .	??





## Chapter 3

# Class Index

### 3.1 Class List

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

Array_Class	..	??
Boy		
	All the variables and member functions are declared here	??
Choosy		
	< A type of girl	??
Couple		??
Desperate		
	< inheritance	??
Fill		
	< Fills data in the array	??
Geek		??
Generous		??
Gift		
	<The header file having all the parameters a gift can have . there are three types of gifts	??
Girl		??
hash		??
MakePair		??
Miser		??
NewCouple		??
NewWay		??
Normal		??
Pair		??
secondary		??
Sorted_Class		??
t_days		??



## Chapter 4

# Class Documentation

### 4.1 Array\_Class Class Reference

```
#include <Array_Class.h>
```

#### Public Member Functions

- void **function** ()

#### 4.1.1 Detailed Description

< Boys are stored in the form of an object array. Inheritance is used .

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

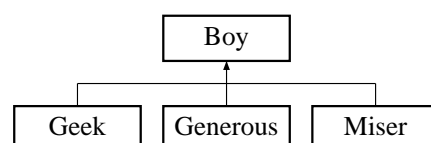
- Array\_Class.h

### 4.2 Boy Class Reference

All the variables and member functions are declared here.

```
#include <boy.h>
```

Inheritance diagram for Boy:



## Public Member Functions

- void `happiness` (int total\_cost, `Girl` &g)

## Public Attributes

- int `committed`
- int `type`
- char `name` [10]
- int `attractive`
- int `intell_b`
- int `budget`
- int `min_attr`
- int `happy`
- int `not_choose`

### 4.2.1 Detailed Description

All the variables and member functions are declared here.

### 4.2.2 Member Function Documentation

#### 4.2.2.1 happiness()

```
void Boy::happiness (  
    int total_cost,  
    Girl & g )
```

<miser

<generous

<geek

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

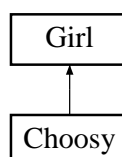
- boy.h
- boy.cpp

## 4.3 Choosy Class Reference

< A type of girl

```
#include <girl_choosy.h>
```

Inheritance diagram for Choosy:



### Public Member Functions

- [Choosy](#) (char \*namei, int atr, int intl, int man\_cost, int criteria)  
< *Constructor*

### Additional Inherited Members

#### 4.3.1 Detailed Description

< A type of girl

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

- girl\_choosy.h

## 4.4 Couple Class Reference

### Public Member Functions

- void [happiness](#) ()
- void [compatibility](#) ()

### Public Attributes

- [Girl](#) **g**
- [Boy](#) **b**
- int **happy**
- int **compatible**

#### 4.4.1 Member Function Documentation

##### 4.4.1.1 compatibility()

```
void Couple::compatibility ( )
```

<Calculates couple's compatibility

##### 4.4.1.2 happiness()

```
void Couple::happiness ( )
```

<Calculates Happiness of the couple

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

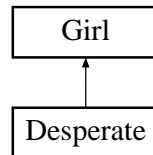
- couple.h
- couple.cpp

## 4.5 Desperate Class Reference

< inheritance

```
#include <girl_desperate.h>
```

Inheritance diagram for Desperate:



### Public Member Functions

- **Desperate** (char \*namei, int atr, int intl, int man\_cost, int criteria)

### Additional Inherited Members

#### 4.5.1 Detailed Description

< inheritance

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

- girl\_desperate.h

## 4.6 Fill Class Reference

< Fills data in the array

```
#include <Fill.h>
```

### Public Member Functions

- void **fill\_data** (Boy b[], Girl go[])  
*will result in transferring information of all girls and boys into array*

### Public Attributes

- int **num\_boys**  
*Number of boys.*
- int **num\_girls**  
*Number of girls.*
- int **num\_gifts**  
*Number of gifts.*

### 4.6.1 Detailed Description

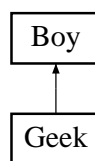
< Fills data in the array

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

- Fill.h
- Fill.cpp

## 4.7 Geek Class Reference

Inheritance diagram for Geek:



### Public Member Functions

- [Geek](#) (char \*namei, int atr, int intl, int budget1, int min\_atr)  
< *Inheritance shown.*

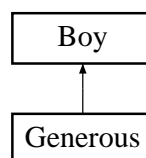
### Additional Inherited Members

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

- boy\_geeks.h

## 4.8 Generous Class Reference

Inheritance diagram for Generous:



### Public Member Functions

- [Generous](#) (char \*namei, int atr, int intl, int budget1, int min\_atr)  
< *inheritance shown.*

## Additional Inherited Members

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

- boy\_generous.h

## 4.9 Gift Class Reference

<The header file having all the parameters a gift can have . there are three types of gifts

```
#include <gift.h>
```

### Public Attributes

- int **value**
- int **price**
- int **type**
- int **which**
- int **luxury\_rate**
- int **difficulty**
- int **utility\_value**
- int **utility\_class**

### 4.9.1 Detailed Description

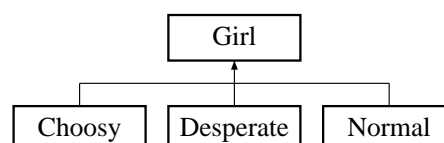
<The header file having all the parameters a gift can have . there are three types of gifts

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

- gift.h

## 4.10 Girl Class Reference

Inheritance diagram for Girl:



### Public Member Functions

- void **happiness** (int total\_cost, int total\_value)



### Public Attributes

- int **committed**
- int **type**
- char **name** [10]
- int **attractive**
- int **intell\_g**
- int **maint**
- int **happy**
- int **check**
- char **no** [10]

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

- girl.h
- girl.cpp

## 4.11 hash Class Reference

### Public Member Functions

- void **function** ()

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

- hash.h
- hash.cpp

## 4.12 MakePair Class Reference

### Public Member Functions

- void **input** ()
- void **find\_happiness** ()  
*k happiest couples found.*
- void **print** ()  
*printing the output in the file.*
- void **giftin** ()  
*gifting done on the basis of the boy's budget.*
- void **break\_up\_function** ()  
*k least happy couples break up*
- void **form\_couple** ()  
*The couples which break up find a suitable partner for themselves if possible.*

## Public Attributes

- [Fill](#) **fobj**
- [Boy](#) **b** [100]
- [Girl](#) **go** [100]
- **int k**
- [Couple](#) **c** [30]
- [Gift](#) **cg** [100][100]

## 4.12.1 Member Function Documentation

### 4.12.1.1 `break_up_function()`

```
void MakePair::break_up_function ( )
```

k least happy couples break up

Happiness of all couples is calculated and couple who are less happy breakup and find a new partner from the remaining single people.

### 4.12.1.2 `giftin()`

```
void MakePair::giftin ( )
```

gifting done on the basis of the boy's budget.

k happy couples are seen.

### 4.12.1.3 `input()`

```
void MakePair::input ( )
```

In this function gifting is done on the basis of boy's budget. Exception handling used in this function if the number of gifts is 0. <The criteria of all the girls in choosing a boy

### 4.12.1.4 `print()`

```
void MakePair::print ( )
```

printing the output in the file.

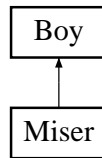
Break Up of k less happy Couples is performed in the `break_up_function`.

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

- `inh_pair.h`
- `inh_pair.cpp`

## 4.13 Miser Class Reference

Inheritance diagram for Miser:



### Public Member Functions

- [Miser](#) (char \*namei, int atr, int intl, int budget1, int min\_atr)  
< Inheritance shown

### Additional Inherited Members

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

- boy\_miser.h

## 4.14 NewCouple Class Reference

```
#include <new_order_couple.h>
```

### Public Member Functions

- void **select** ()

### 4.14.1 Detailed Description

< Output stored in order\_couple.txt Exception Handled when the number of boys and number of girls in the text file is 0.

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

- new\_order\_couple.h

## 4.15 NewWay Class Reference

### Public Member Functions

- void [allocation](#) ()  
*This function is the second type of gift allocation in which every type of gift is given then the normal distribution is done according to the budget left.*

## Public Attributes

- int **k**
- Couple **c** [30]

### 4.15.1 Member Function Documentation

#### 4.15.1.1 allocation()

```
void NewWay::allocation ( )
```

This function is the second type of gift allocation in which every type of gift is given then the normal distribution is done according to the budget left.

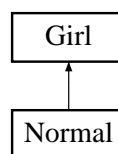
< Output in happ\_comp\_gift

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

- NewWay.h
- NewWay.cpp

## 4.16 Normal Class Reference

Inheritance diagram for Normal:



## Public Member Functions

- **Normal** (char \*namei, int atr, int intl, int man\_cost, int criteria)

## Additional Inherited Members

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

- girl\_normal.h

## 4.17 Pair Class Reference

### Public Member Functions

- void `input` ()  
    *< Exception Handling used for the case if the number of couples is 0 or there are no gifts .*
- void `find_happiness` ()

### Public Attributes

- int `k`
- `Couple` `c` [30]
- `Gift` `cg` [100][100]

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

- `make_pair.h`
- `make_pair.cpp`

## 4.18 secondary Class Reference

### Public Member Functions

- void `newList` ()
- void `formPair` ()
- void `newGifts` ()
- void `gifting` ()

### Public Attributes

- `Gift` `gf` [40]
- `Girl` `g` [20]
- `Boy` `b` [31]
- `Couple` `c` [30]
- int `k`
- int `total`

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

- `secondary.h`
- `secondary.cpp`

## 4.19 Sorted\_Class Class Reference

### Public Member Functions

- void **function** ()

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

- Sorted\_Class.h
- Sorted\_Class.cpp

## 4.20 t\_days Class Reference

### Public Member Functions

- void **break\_up\_function** (int t)
- void **print** ()

### Public Attributes

- [MakePair](#) **br**

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

- t\_days.h
- t\_days.cpp