

# **My Project**

AUTHOR  
Version  
10/04/2017



# Table of Contents

Table of contents

---

# Class Index

## Class List

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

|  |
|--|
| <a href="#">Boys</a> ( <a href="#">Boys</a> class ) .....                    |
| <a href="#">breakup_couple</a> (Contain breakup details ) .....              |
| <a href="#">Couple</a> ( <a href="#">Couple</a> class to make couple ) ..... |
| <a href="#">csv_gen</a> (Class to generate csv file ) .....                  |
| <a href="#">gift</a> (Gift class ) .....                                     |
| <a href="#">Girls</a> ( <a href="#">Girls</a> Class ) .....                  |
| <a href="#">MyLogger</a> (Class <a href="#">MyLogger</a> ) .....             |
| <a href="#">q4</a> (Main class <a href="#">q4</a> ) .....                    |

---

# Class Documentation

## Boys Class Reference

[Boys](#) class.

---

### Detailed Description

[Boys](#) class.

---

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

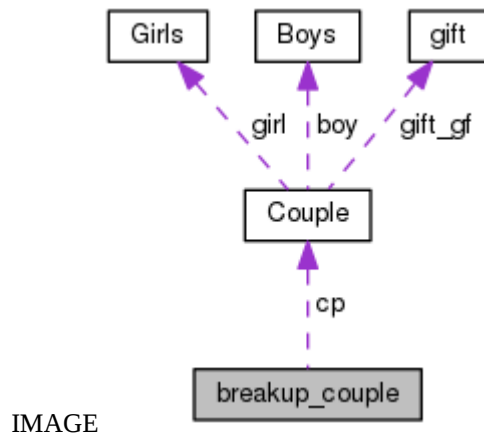
1 Boys.java

---

## breakup\_couple Class Reference

Contain breakup details.

Collaboration diagram for breakup\_couple:



IMAGE

---

### Detailed Description

Contain breakup details.

---

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

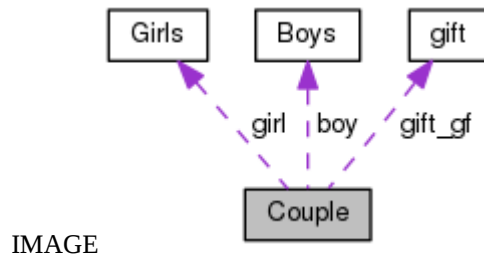
2 breakup\_couple.java

---

## Couple Class Reference

[Couple](#) class to make couple.

Collaboration diagram for Couple:



IMAGE

---

### Detailed Description

[Couple](#) class to make couple.

---

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

3 Couple.java

---

## csv\_gen Class Reference

Class to generate csv file.

---

### Detailed Description

Class to generate csv file.

---

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

4 csv\_gen.java

---

## gift Class Reference

gift class

---

### Detailed Description

gift class

---

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

5 gift.java

---

## Girls Class Reference

[Girls](#) Class.

---

### Detailed Description

[Girls](#) Class.

---

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

6 Girls.java

---

## MyLogger Class Reference

Class [MyLogger](#).

---

### Detailed Description

Class [MyLogger](#).

---

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

7 MyLogger.java

---

## q4 Class Reference

Main class [q4](#).

## Static Public Member Functions

static void [main](#) (String args[]) throws IOException

---

## Detailed Description

Main class [q4](#).

---

## Member Function Documentation

**static void q4.main (String args[]) throws IOException[static]**

Sorting Gift according to price

```
8                                     {
9         int i,j;
10        i=0;
11        j=0;
12        int m=0;
13        Boys b[] = new Boys[1000];
14        csv\_gen csv_files = new csv\_gen();
15        csv_files.csv();
16        String csvFile = "boy.csv";
17        File out=new File("log.log");
18        String line = "";
19        String csvSplit = ",";
20        BufferedReader buff = null;
21        try{
22            buff = new BufferedReader(new FileReader(csvFile));
23            while((line = buff.readLine() )!= null){
24                String[] boys_table = line.split(csvSplit);
25
26                b[i] = new Boys();
27                b[i].name = boys_table[0];
28                b[i].intelligence = Integer.parseInt(boys_table[1]);
29                b[i].attractivness = Integer.parseInt(boys_table[2]);
30                b[i].min_attr_req = Integer.parseInt(boys_table[3]);
31                b[i].budget = Integer.parseInt(boys_table[4]);
32                b[i].status = boys_table[5];
33                b[i].boy_type= boys_table[6];
34                b[i].girlf="";
35                b[i].happiness_boy = 0;
36                i++;
37            }
38        }
39
40        catch(FileNotFoundException e){
41            e.printStackTrace();
42        }catch(IOException e){
43            e.printStackTrace();
44        }
45
46        Girls g[] = new Girls[1000];
47        csvFile = "girl.csv";
48        line = "";
49        csvSplit = ",";
50        buff = null;
51
52
53        try{
54            buff = new BufferedReader(new FileReader(csvFile));
55            while((line = buff.readLine() )!= null){
56                String[] girls_table = line.split(csvSplit);
57
58                g[j] = new Girls();
59                g[j].name = girls_table[0];
60                g[j].attar = Integer.parseInt(girls_table[1]);
61                g[j].expense = Integer.parseInt(girls_table[2]);
62                g[j].intelligence = Integer.parseInt(girls_table[3]);
63                g[j].status = girls_table[4];
```

```

64         g[j].girl_type = girls_table[5];
65         g[j].boyf = "";
66         g[j].happiness_girl= 0;
67     }
68     j++;
69 }
70 }
71 catch(FileNotFoundException e){
72     e.printStackTrace();
73 }catch(IOException e){
74     e.printStackTrace();
75 }finally{
76     if(buff != null){
77         try{
78             buff.close();
79         }catch(IOException e){
80             e.printStackTrace();
81         }
82     }
83 }
84
85
86 gift gf[] = new gift[1000];
87 csvFile = "gift.csv";
88 line = "";
89 csvSplit = ",";
90 buff = null;
91
92
93 try{
94     buff = new BufferedReader(new FileReader(csvFile));
95     while((line = buff.readLine()) != null){
96         String[] gifts_table = line.split(csvSplit);
97         // System.out.println(boy[0] + boy[1] + boy[2] + boy [3]);
98         gf[m] = new gift();
99         gf[m].gift_name = gifts_table[0];
100        gf[m].gift_price = Integer.parseInt(gifts_table[1]);
101
102        gf[m].gift_value = Integer.parseInt(gifts_table[2]);
103        gf[m].gift_type = gifts_table[3];
104
105        m++;
106    }
107 }
108 catch(FileNotFoundException e){
109     e.printStackTrace();
110 }catch(IOException e){
111     e.printStackTrace();
112 }finally{
113     if(buff != null){
114         try{
115             buff.close();
116         }catch(IOException e){
117             e.printStackTrace();
118         }
119     }
120 }
121
122
123 gift tem;
124 int m1,n;
125 for(m1=0;m1<m;m1++){
126     for(n=m1+1;n<m;n++){
127         if(gf[m1].gift_price > gf[n].gift_price){
128             tem= gf[m1];
129             gf[m1]=gf[n];
130             gf[n]=tem;
131         }
132     }
133 }
134
135
136
137
138 int count = 0;
139 String output = "";
140 int c = 0,d = 0;
141 Couple gbccouple[] = new Couple[100];

```



```

142         for(c=0;c<i;c++){
143             for(d=0;d<j;d++){
144                 if(b[c].is_eligible(g[d].expense,g[d].attar) &&
b[c].status.equals("Single") && g[d].status.equals("Single")){
145                     b[c].girlf = g[d].name;
146                     g[d].boyf = b[c].name;
147                     b[c].status = "Is_committed";
148                     g[d].status = "Is_committed";
149                     count++;
150                     String df= new SimpleDateFormat("dd/MM/yy
HH:mm:ss").format(new Time(System.currentTimeMillis()));
151                     output = df+" "+"Boy : "+b[c].name + " is Committed
with " +"Girl : "+g[d].name + "\n" ;
152                     System.out.println(output);
153                     try(BufferedWriter buffer=new BufferedWriter(new
FileWriter(out,true))){
154                         buffer.write(output);
155                     }
156                     gbcouple[count-1] = new Couple(b[c],g[d]);
157                 }
158             }
159         }
160     }
161 }
162 }
163 int o;
164
165 Random rn = new Random();
166 o = rn.nextInt(count);
167 breakup_couple bk = new breakup_couple(gbcouple, count, o,output);//
Least n happy couples broke up
168 String outp = "";
169 outp = bk.break_print();
170 for(i =0 ; i < o; i++)
171 {
172     for(j = 0; j < 20; j++)
173     {
174         if(b[j].status == "Single" && gbcouple[i].girl.status ==
"Single"&& !((b[j].name).equals(gbcouple[i].boy.name)) && b[j].budget >=
gbcouple[i].girl.expense && gbcouple[i].girl.attar >= b[j].min_attr_req) {
175             b[j].status = "Is_committed";
176             gbcouple[i].girl.status = "Is_committed";
177             String fd= new SimpleDateFormat("dd/MM/yy
HH:mm:ss").format(new Time(System.currentTimeMillis()));
178             outp = outp.concat("\n"+fd+ " "+"New Commitment :
"+gbcouple[i].girl.name + " and "+b[j].name+" \n\n");//New commitments of broke
up girls
179             // System.out.println("New Commitment after break up:
"+gbcouple[i].girl.name + "-" +b[j].name );
180             gbcouple[i].boy = b[j];
181             break;
182         }
183     }
184 }
185 }
186 }
187 System.out.println(outp);
188 try(BufferedWriter buffer=new BufferedWriter(new
FileWriter(out,true))){
189     buffer.write(outp);
190 }
191 }
192 }

```

---

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

8 q4.java

---

## Index

INDE

