

# Assignment PPL Question-5

API Documentation

April 10, 2017

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Module Choosy</b>	<b>2</b>
1.1 Variables . . . . .	2
1.2 Class Choosy . . . . .	2
1.2.1 Methods . . . . .	2
1.2.2 Class Variables . . . . .	2
<b>2 Module Desperate</b>	<b>3</b>
2.1 Variables . . . . .	3
2.2 Class Desperate . . . . .	3
2.2.1 Methods . . . . .	3
2.2.2 Class Variables . . . . .	3
<b>3 Module Essential</b>	<b>4</b>
3.1 Variables . . . . .	4
3.2 Class Essential . . . . .	4
3.2.1 Methods . . . . .	4
3.2.2 Class Variables . . . . .	4
<b>4 Module Geek</b>	<b>5</b>
4.1 Variables . . . . .	5
4.2 Class Geek . . . . .	5
4.2.1 Methods . . . . .	5
4.2.2 Class Variables . . . . .	5
<b>5 Module Generous</b>	<b>6</b>
5.1 Variables . . . . .	6
5.2 Class Generous . . . . .	6
5.2.1 Methods . . . . .	6
5.2.2 Class Variables . . . . .	6
<b>6 Module Luxury</b>	<b>7</b>
6.1 Variables . . . . .	7
6.2 Class Luxury . . . . .	7
6.2.1 Methods . . . . .	7
6.2.2 Class Variables . . . . .	7

<b>7</b>	<b>Module Miser</b>	<b>8</b>
7.1	Variables . . . . .	8
7.2	Class Miser . . . . .	8
7.2.1	Methods . . . . .	8
7.2.2	Class Variables . . . . .	8
<b>8</b>	<b>Module Normal</b>	<b>9</b>
8.1	Variables . . . . .	9
8.2	Class Normal . . . . .	9
8.2.1	Methods . . . . .	9
8.2.2	Class Variables . . . . .	9
<b>9</b>	<b>Module Utility</b>	<b>10</b>
9.1	Variables . . . . .	10
9.2	Class Utility . . . . .	10
9.2.1	Methods . . . . .	10
9.2.2	Class Variables . . . . .	10
<b>10</b>	<b>Module boy</b>	<b>11</b>
10.1	Variables . . . . .	11
10.2	Class Boy . . . . .	11
10.2.1	Methods . . . . .	11
10.2.2	Class Variables . . . . .	11
<b>11</b>	<b>Module couples</b>	<b>13</b>
11.1	Variables . . . . .	13
11.2	Class Couple . . . . .	13
11.2.1	Methods . . . . .	13
<b>12</b>	<b>Module gift_item</b>	<b>14</b>
12.1	Variables . . . . .	14
12.2	Class Gift . . . . .	14
12.2.1	Methods . . . . .	14
12.2.2	Class Variables . . . . .	14
<b>13</b>	<b>Module girl</b>	<b>15</b>
13.1	Variables . . . . .	15
13.2	Class girl . . . . .	15
13.2.1	Methods . . . . .	15
13.2.2	Class Variables . . . . .	15
<b>14</b>	<b>Module logs</b>	<b>17</b>
14.1	Functions . . . . .	17
14.2	Variables . . . . .	17
<b>15</b>	<b>Module main</b>	<b>18</b>
15.1	Functions . . . . .	18
<b>16</b>	<b>Module util</b>	<b>19</b>
16.1	Functions . . . . .	19
16.2	Variables . . . . .	19

# 1 Module Choosy

## 1.1 Variables

Name	Description
<code>__package__</code>	Value: None

## 1.2 Class Choosy



Class for choosy girl

### 1.2.1 Methods

<code>__init__(self, name, attractiveness, maintenance_budget, intelligence, type_)</code>
Constructor for initialising choosy girl
Overrides: <code>girl.girl.__init__</code>

*Inherited from `girl.girl` (Section 13.2)*

`is_eligible()`, `modify_maintenance_budget()`, `set_boyfriend()`, `set_happiness()`

### 1.2.2 Class Variables

Name	Description
<i>Inherited from <code>girl.girl</code> (Section 13.2)</i>	
<code>attractiveness</code> , <code>boyfriend</code> , <code>happiness</code> , <code>intelligence</code> , <code>maintenance_budget</code> , <code>name</code> , <code>relationship_status</code> , <code>type_</code>	

## 2 Module Desperate

### 2.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 2.2 Class Desperate

girl.girl —  
Desperate.Desperate

Class for desperate girls

#### 2.2.1 Methods

<code>__init__(self, name, attractiveness, maintenance_budget, intelligence, type_)</code>
Constructor for initialising Desperate girl
Overrides: girl.girl.__init__

*Inherited from girl.girl (Section 13.2)*

`is_eligible()`, `modify_maintenance_budget()`, `set_boyfriend()`, `set_happiness()`

#### 2.2.2 Class Variables

Name	Description
<i>Inherited from girl.girl (Section 13.2)</i>	
<code>attractiveness</code> , <code>boyfriend</code> , <code>happiness</code> , <code>intelligence</code> , <code>maintenance_budget</code> , <code>name</code> , <code>relationship_status</code> , <code>type_</code>	

### 3 Module Essential

#### 3.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

#### 3.2 Class Essential

gift\_item.Gift —  
**Essential.Essential**

Class for Essential gift

##### 3.2.1 Methods

<b><code>__init__</code></b> ( <i>self, name, cost, value, type__</i> )
Constructor for initialising essential gifts.
Overrides: gift_item.Gift. <code>__init__</code>

##### 3.2.2 Class Variables

Name	Description
<i>Inherited from gift_item.Gift (Section 12.2)</i>	
cost, name, type__, value	

## 4 Module Geek

### 4.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 4.2 Class Geek

boy.Boy —  
**Geek.Geek**

Class for geek boy

#### 4.2.1 Methods

<code>__init__</code> ( <i>self, name, attractiveness, girlfriend_budget, intelligence, minimum_attraction_requirement, type_</i> )
Constructor for initialising geek boy
Overrides: boy.Boy. <code>__init__</code>

*Inherited from boy.Boy (Section 10.2)*

`is_eligible()`, `modify_boys_girlfriend_budget()`, `set_girlfriend()`, `set_happiness()`

#### 4.2.2 Class Variables

Name	Description
<i>Inherited from boy.Boy (Section 10.2)</i>	
attractiveness, boys_girlfriend_budget, girlfriend, happiness, intelligence, minimum_attraction_requirement, name, relationship_status, type_	

## 5 Module Generous

### 5.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 5.2 Class Generous

boy.Boy —  
**Generous.Generous**

Class for generous boy

#### 5.2.1 Methods

<code>__init__</code> ( <i>self, name, attractiveness, girlfriend_budget, intelligence, minimum_attraction_requirement, type_</i> )
Constructor for initialising generous boy
Overrides: boy.Boy. <code>__init__</code>

*Inherited from boy.Boy (Section 10.2)*

`is_eligible()`, `modify_boys_girlfriend_budget()`, `set_girlfriend()`, `set_happiness()`

#### 5.2.2 Class Variables

Name	Description
<i>Inherited from boy.Boy (Section 10.2)</i>	
<code>attractiveness</code> , <code>boys_girlfriend_budget</code> , <code>girlfriend</code> , <code>happiness</code> , <code>intelligence</code> , <code>minimum_attraction_requirement</code> , <code>name</code> , <code>relationship_status</code> , <code>type_</code>	

## 6 Module *Luxury*

### 6.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 6.2 Class *Luxury*

`gift_item.Gift` —  
                     ***Luxury.Luxury***

Class for *Luxury* gift

#### 6.2.1 Methods

<code>__init__(self, name, cost, value, type__)</code>
Constructor for initialising the <i>Luxury</i> gift
Overrides: <code>gift_item.Gift.__init__</code>

#### 6.2.2 Class Variables

Name	Description
<i>Inherited from <code>gift_item.Gift</code> (Section 12.2)</i>	
<code>cost</code> , <code>name</code> , <code>type__</code> , <code>value</code>	



## 7 Module Miser

### 7.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 7.2 Class Miser

```
boy.Boy └─ Miser.Miser
```

Class for miser boy

#### 7.2.1 Methods

<code>__init__</code> ( <i>self, name, attractiveness, girlfriend_budget, intelligence, minimum_attraction_requirement, type_</i> )
Constructor for initialising Miser boy
Overrides: boy.Boy. <code>__init__</code>

*Inherited from boy.Boy (Section 10.2)*

`is_eligible()`, `modify_boys_girlfriend_budget()`, `set_girlfriend()`, `set_happiness()`

#### 7.2.2 Class Variables

Name	Description
<i>Inherited from boy.Boy (Section 10.2)</i>	
<code>attractiveness</code> , <code>boys_girlfriend_budget</code> , <code>girlfriend</code> , <code>happiness</code> , <code>intelligence</code> , <code>minimum_attraction_requirement</code> , <code>name</code> , <code>relationship_status</code> , <code>type_</code>	

## 8 Module Normal

### 8.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 8.2 Class Normal

girl.girl —  
**Normal.Normal**

Class for normal girl

#### 8.2.1 Methods

<code>__init__</code> ( <i>self</i> , <i>name</i> , <i>attractiveness</i> , <i>maintenance_budget</i> , <i>intelligence</i> , <i>type_</i> )
Constructor for initialising Normal girl.
Overrides: girl.girl. <code>__init__</code>

*Inherited from girl.girl (Section 13.2)*

`is_eligible()`, `modify_maintenance_budget()`, `set_boyfriend()`, `set_happiness()`

#### 8.2.2 Class Variables

Name	Description
<i>Inherited from girl.girl (Section 13.2)</i>	
attractiveness, boyfriend, happiness, intelligence, maintenance_budget, name, relationship_status, type_	

## 9 Module Utility

### 9.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 9.2 Class Utility

`gift_item.Gift` —  
                     **Utility.Utility**

Class for utility gift

#### 9.2.1 Methods

<code>__init__(self, name, cost, value, type_)</code>
Constructor for initialising Utility gift.
Overrides: <code>gift_item.Gift.__init__</code>

#### 9.2.2 Class Variables

Name	Description
<i>Inherited from <code>gift_item.Gift</code> (Section 12.2)</i>	
<code>cost, name, type_, value</code>	

## 10 Module boy

### 10.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 10.2 Class Boy

**Known Subclasses:** Geek.Geek, Generous.Generous, Miser.Miser

Class for BOY

#### 10.2.1 Methods

<code>__init__(self, name, attractiveness, girlfriend_budget, intelligence, minimum_attraction_requirement, type_)</code>
Constructor for initialising boy class
<code>set_happiness(self, happiness)</code>
Method for setting happiness
<code>set_girlfriend(self, girlfriend)</code>
Method for setting girlfriend
<code>modify_boys_girlfriend_budget(self, budget)</code>
Method for modifying the girlfriend budget
<code>is_eligible(self, maintenance_budget, attractiveness)</code>
Method for checking the eligibility

#### 10.2.2 Class Variables

Name	Description
name	<b>Value:</b> ''
attractiveness	<b>Value:</b> 0
boys_girlfriend_budget	<b>Value:</b> 0

*continued on next page*

Name	Description
intelligence	Value: 0
minimum_attraction_requirement	Value: 0
type__	Value: ''
relationship_status	Value: 'single'
happiness	Value: 0
girlfriend	Value: ''

## 11 Module couples

### 11.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 11.2 Class Couple

Class for couples

#### 11.2.1 Methods

<code>__init__(self, boy, girl)</code>
Constructor for intialising couple

<code>set_compatibility(self)</code>
Method for setting compatibility

<code>set_happiness(self)</code>
Method for setting Happiness

## 12 Module gift\_item

### 12.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 12.2 Class Gift

**Known Subclasses:** Essential.Essential, Luxury.Luxury, Utility.Utility

#### 12.2.1 Methods

<code>__init__(self, name, cost, value, type_)</code>
Constructor for initialising gift

#### 12.2.2 Class Variables

Name	Description
name	<b>Value:</b> ''
cost	<b>Value:</b> 0
value	<b>Value:</b> 0
type_	Class for gift <b>Value:</b> ''

## 13 Module girl

### 13.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

### 13.2 Class girl

**Known Subclasses:** Choosy.Choosy, Desperate.Desperate, Normal.Normal

Class for girl

#### 13.2.1 Methods

<code>__init__(self, name, attractiveness, maintenance_budget, intelligence, type_)</code>
Constructor fot initialising girl
<code>set_happiness(self, happiness)</code>
Method for setting happiness
<code>set_boyfriend(self, boyfriend)</code>
Fucntion for setting boyfriend
<code>modify_maintenance_budget(self, budget)</code>
Function for modifying maintainence budget
<code>is_eligible(self, boys_girlfriend_budget)</code>
Method for checking the eligibility

#### 13.2.2 Class Variables

Name	Description
name	<b>Value:</b> ''
attractiveness	<b>Value:</b> 0
maintenance_budget	<b>Value:</b> 0

*continued on next page*



Name	Description
intelligence	Value: 0
relationship_status	Value: ''
boyfriend	Value: ''
happiness	Value: 0
type_	Value: ''

## 14 Module logs

### 14.1 Functions

<code>log_maker(<i>write</i>)</code>
Function for creating log file

### 14.2 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

## 15 Module main

### 15.1 Functions

<b>calc_happiness</b> ( $H$ )
Function to calculate happiness

<b>happy_couple</b> ( $H, k$ )
Function for calculating k happy couples

<b>generous</b> ( $gifts, p$ )
Function for generous boy

<b>miser</b> ( $gifts, x$ )
Function for miser boy

<b>geek</b> ( $gifts, c$ )
Function for geek boy

<b>gifts_details</b> ( $H$ )
Function for gift details

<b>test</b> ()
Function to test all other functions.

## 16 Module util

### 16.1 Functions

<b>testing_util()</b>
Function for creating random girls, boys and gifts.

<b>create(<i>file_name</i>, <i>list_name</i>)</b>
Function to open a file and write content in it.

### 16.2 Variables

Name	Description
__package__	<b>Value:</b> None

## Index

- boy (*module*), 11–12
  - boy.Boy (*class*), 11–12
    - boy.Boy.\_\_init\_\_ (*method*), 11
    - boy.Boy.is\_eligible (*method*), 11
    - boy.Boy.modify\_boys\_girlfriend\_budget (*method*), 11
    - boy.Boy.set\_girlfriend (*method*), 11
    - boy.Boy.set\_happiness (*method*), 11
- Choosy (*module*), 2
  - Choosy.Choosy (*class*), 2
- couples (*module*), 13
  - couples.Couple (*class*), 13
    - couples.Couple.\_\_init\_\_ (*method*), 13
    - couples.Couple.set\_compatibility (*method*), 13
    - couples.Couple.set\_happiness (*method*), 13
- Desperate (*module*), 3
  - Desperate.Desperate (*class*), 3
- Essential (*module*), 4
  - Essential.Essential (*class*), 4
- Geek (*module*), 5
  - Geek.Geek (*class*), 5
- Generous (*module*), 6
  - Generous.Generous (*class*), 6
- gift\_item (*module*), 14
  - gift\_item.Gift (*class*), 14
    - gift\_item.Gift.\_\_init\_\_ (*method*), 14
- girl (*module*), 15–16
  - girl.girl (*class*), 15–16
    - girl.girl.\_\_init\_\_ (*method*), 15
    - girl.girl.is\_eligible (*method*), 15
    - girl.girl.modify\_maintenance\_budget (*method*), 15
    - girl.girl.set\_boyfriend (*method*), 15
    - girl.girl.set\_happiness (*method*), 15
- logs (*module*), 17
  - logs.log\_maker (*function*), 17
- Luxury (*module*), 7
  - Luxury.Luxury (*class*), 7
- main (*module*), 18
  - main.calc\_happiness (*function*), 18
  - main.geek (*function*), 18
  - main.generous (*function*), 18
  - main.gifts\_details (*function*), 18
  - main.happy\_couple (*function*), 18
  - main.miser (*function*), 18
  - main.test (*function*), 18
- Miser (*module*), 8
  - Miser.Miser (*class*), 8
- Normal (*module*), 9
  - Normal.Normal (*class*), 9
- util (*module*), 19
  - util.create (*function*), 19
  - util.testing\_util (*function*), 19
- Utility (*module*), 10
  - Utility.Utility (*class*), 10