

Darwin

Generated by Doxygen 1.9.3



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 Individual Class Reference	5
3.1.1 Constructor & Destructor Documentation	6
3.1.1.1 Individual()	6
3.1.2 Member Function Documentation	6
3.1.2.1 fitness_check()	6
3.1.3 Friends And Related Function Documentation	7
3.1.3.1 operator<< [1/2]	7
3.1.3.2 operator<< [2/2]	7
3.1.4 Member Data Documentation	7
3.1.4.1 chromosomes	7
3.2 Population Class Reference	8
3.2.1 Detailed Description	9
3.2.2 Constructor & Destructor Documentation	9
3.2.2.1 Population()	9
3.2.3 Member Function Documentation	9
3.2.3.1 crossover()	9
3.2.3.2 fitnesscheckpopulation()	10
3.2.3.3 randomInteger()	10
3.2.3.4 read_population_from_file()	10
3.2.3.5 writeToFile()	10
3.2.4 Member Data Documentation	11
3.2.4.1 population	11
<b>4 File Documentation</b>	<b>13</b>
4.1 Individual.cpp File Reference	13
4.1.1 Detailed Description	13
4.1.2 Function Documentation	14
4.1.2.1 operator<<()	14
4.2 Individual.h File Reference	14
4.2.1 Detailed Description	15
4.3 Individual.h	15
4.4 main.cpp File Reference	16
4.4.1 Detailed Description	16
4.5 Population.cpp File Reference	17
4.5.1 Detailed Description	17
4.6 Population.h File Reference	18

4.6.1 Detailed Description . . . . .	18
4.7 Population.h . . . . .	19
<b>Index</b>	<b>21</b>

# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">Individual</a>	.....	<a href="#">5</a>
<a href="#">Population</a>	.....	<a href="#">8</a>



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">Individual.cpp</a>	13
<a href="#">Individual.h</a>	14
<a href="#">main.cpp</a>	16
<a href="#">Population.cpp</a>	17
<a href="#">Population.h</a>	18





## Chapter 3

# Class Documentation

### 3.1 Individual Class Reference

Collaboration diagram for Individual:

Individual
+ vector< int > chromosomes
+ Individual(const vector< int > &_chrs) + Individual(vector< int > &&_chrs) + Individual(const Individual &other)=default + Individual(Individual &&other)=default + Individual()=default + Individual & operator=(Individual &&other)=default + ~Individual() + double fitness_check()

#### Public Member Functions

- [Individual](#) (const vector< int > &\_chrs)
- **Individual** (vector< int > &&\_chrs)
- **Individual** (const [Individual](#) &other)=default
- **Individual** ([Individual](#) &&other)=default
- [Individual](#) & **operator=** ([Individual](#) &&other)=default
- double [fitness\\_check](#) ()

## Public Attributes

- vector< int > [chromosomes](#)

## Friends

- ostream & [operator<<](#) (ostream &os, const [Individual](#) &ind)
- ostream & [operator<<](#) (ostream &os, const [Individual](#) &ind)

## 3.1.1 Constructor & Destructor Documentation

### 3.1.1.1 Individual()

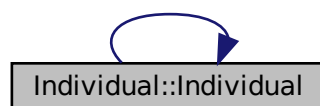
```
Individual::Individual (
    const vector< int > &_chrs ) [explicit]
```

Constructor that assigns chromosomes to individuals .

#### Parameters

<code>_chrs</code>	chromosomes
--------------------	-------------

Here is the call graph for this function:



## 3.1.2 Member Function Documentation

### 3.1.2.1 fitness\_check()

```
double Individual::fitness_check ( )
```

Function that does the fitness check for each individuals

### 3.1.3 Friends And Related Function Documentation

#### 3.1.3.1 `operator<<` [1/2]

```
ostream & operator<< (
    ostream & os,
    const Individual & ind ) [friend]
```

output overload operator

##### Parameters

<i>os</i>	output stream
<i>individual</i>	object of class <a href="#">Individual</a>

#### 3.1.3.2 `operator<<` [2/2]

```
ostream & operator<< (
    ostream & os,
    const Individual & ind ) [friend]
```

output overload operator

##### Parameters

<i>ostream</i>	
<i>os</i>	output stream
<i>individual</i>	object

output overload operator

##### Parameters

<i>os</i>	output stream
<i>individual</i>	object of class <a href="#">Individual</a>

### 3.1.4 Member Data Documentation

#### 3.1.4.1 `chromosomes`

```
vector<int> Individual::chromosomes
```

Vector of chromosomes for each [Individual](#)

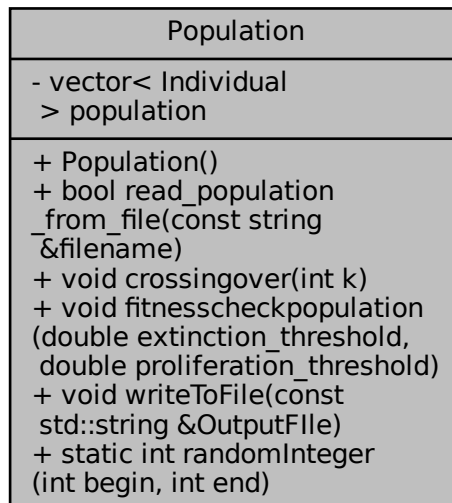
The documentation for this class was generated from the following files:

- [Individual.h](#)
- [Individual.cpp](#)

## 3.2 Population Class Reference

```
#include <Population.h>
```

Collaboration diagram for Population:



### Public Member Functions

- [Population](#) ()
- bool [read\\_population\\_from\\_file](#) (const string &filename)
- void [crossingover](#) (int k)
- void [fitnesscheckpopulation](#) (double extinction\_threshold, double proliferation\_threshold)
- void [writeToFile](#) (const std::string &OutputFile)

### Static Public Member Functions

- static int [randomInteger](#) (int begin, int end)

## Private Attributes

- vector< [Individual](#) > [population](#)

### 3.2.1 Detailed Description

Class of [Population](#). Contains vector of individuals.

### 3.2.2 Constructor & Destructor Documentation

#### 3.2.2.1 Population()

```
Population::Population ( )
```

Default constructor

### 3.2.3 Member Function Documentation

#### 3.2.3.1 crossover()

```
void Population::crossover (
    int k )
```

Function that crosses over pairs of individual

##### Parameters

$k$	Number of pairs
-----	-----------------

Here is the call graph for this function:



### 3.2.3.2 fitnesscheckpopulation()

```
void Population::fitnesscheckpopulation (
    double extinction_threshold,
    double proliferation_threshold )
```

Function that does the fitness check of [Population](#) after the crossing\_over.

#### Parameters

<i>extinction_threshold</i>	Extinction threshold "w"
<i>proliferation_threshold</i>	Proliferation threshold "r"

### 3.2.3.3 randomInteger()

```
int Population::randomInteger (
    int begin,
    int end ) [static]
```

Function that generates a random variable depending on the range

#### Parameters

<i>begin</i>	beginning of the range
<i>end</i>	The end of the range

### 3.2.3.4 read\_population\_from\_file()

```
bool Population::read_population_from_file (
    const string & filename )
```

Boolean Function that reads individuals from the file and assign them to population vector

#### Parameters

<i>filename</i>	Input file name
-----------------	-----------------

### 3.2.3.5 writeToFile()

```
void Population::writeToFile (
    const std::string & OutputFile )
```

Function that writes the individuals to the file.

## Parameters

<i>OutputFile</i>	Output file name
-------------------	------------------

## 3.2.4 Member Data Documentation

### 3.2.4.1 population

```
vector<Individual> Population::population [private]
```

Vector of individuals

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

- [Population.h](#)
- [Population.cpp](#)





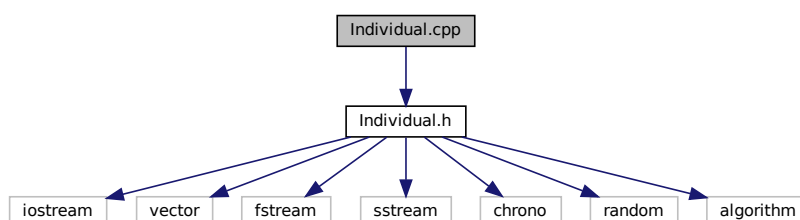
## Chapter 4

# File Documentation

### 4.1 Individual.cpp File Reference

```
#include "Individual.h"
```

Include dependency graph for Individual.cpp:



### Functions

- ostream & [operator<<](#) (ostream &os, const [Individual](#) &ind)

#### 4.1.1 Detailed Description

##### Author

youssef Albali Contains the functions definitions of the [Individual](#) class

##### Version

1.2

##### Date

2022-01-05

##### Copyright

Copyright (c) 2022

## 4.1.2 Function Documentation

### 4.1.2.1 `operator<<()`

```
ostream & operator<< (
    ostream & os,
    const Individual & ind )
```

output overload operator

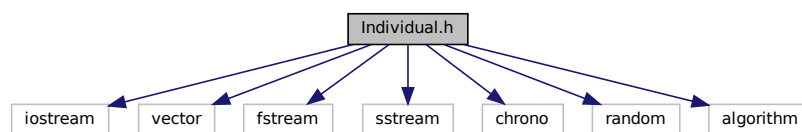
#### Parameters

<i>os</i>	output stream
<i>individual</i>	object of class <a href="#">Individual</a>

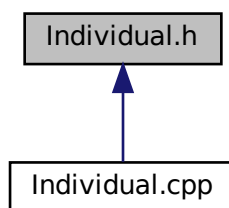
## 4.2 Individual.h File Reference

```
#include <iostream>
#include <vector>
#include <fstream>
#include <sstream>
#include <chrono>
#include <random>
#include <algorithm>
```

Include dependency graph for Individual.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Individual](#)

### 4.2.1 Detailed Description

#### Author

youssef Albali Continues the class [Individual](#) with the declaration of its methods

#### Version

1.2

#### Date

2022-01-05

#### Copyright

Copyright (c) 2022

## 4.3 Individual.h

[Go to the documentation of this file.](#)

```
1
11 #ifndef DARWIN_PROJECT_INDIVIDUAL_H
12 #define DARWIN_PROJECT_INDIVIDUAL_H
13
14 #include <iostream>
15 #include <vector>
16 #include <fstream>
17 #include <sstream>
18 #include <chrono>
19 #include <random>
20 #include <algorithm>
21
22 using namespace std;
```

```

23
24 class Individual {
25
26
27 public:
28     vector<int> chromosomes;
35     explicit Individual(const vector<int> &_chrs);
36
37     Individual(vector<int> &&_chrs);
38
39
40     Individual(const Individual &other) = default;
41     Individual(Individual &&other) = default;
42
43     Individual() = default;
44
45     Individual &operator=(Individual &&other) = default;
46
47     ~Individual()
48     {
49         //only for debugging
50     }
51
52
56     double fitness_check();
57
63     friend ostream &operator<<(ostream &os, const Individual &ind);
64
71     friend ostream &operator<<(ostream &os, const Individual &ind);
72
73 };
74
75 #endif // DARWIN_PROJECT_INDIVIDUAL_H

```

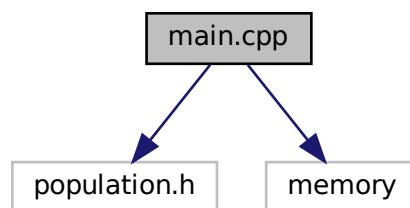
## 4.4 main.cpp File Reference

```

#include "population.h"
#include <memory>

```

Include dependency graph for main.cpp:



## Functions

- void **printHelpMessage** ()
- int **main** (int argc, char \*argv[])

### 4.4.1 Detailed Description

**Author**

youssef Albali Contains one function and the main

**Version**

1.2

**Date**

2022-01-05

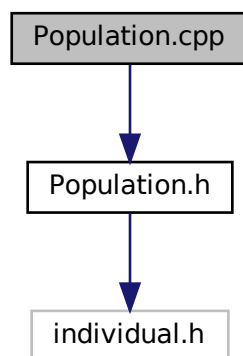
**Copyright**

Copyright (c) 2022

## 4.5 Population.cpp File Reference

```
#include "Population.h"
```

Include dependency graph for Population.cpp:



### 4.5.1 Detailed Description

**Author**

youssef Albali Contains the functions definitions of the [Population](#) class

**Version**

1.2

**Date**

2022-01-05

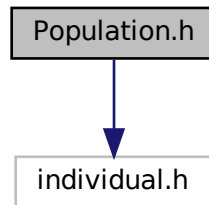
**Copyright**

Copyright (c) 2022

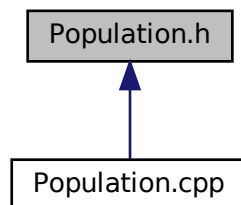
## 4.6 Population.h File Reference

```
#include "individual.h"
```

Include dependency graph for Population.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Population](#)

#### 4.6.1 Detailed Description

##### Author

youssef Albali Contains the class [Population](#) with the declaration of its methods

##### Version

1.2

##### Date

2022-01-05

##### Copyright

Copyright (c) 2022

## 4.7 Population.h

[Go to the documentation of this file.](#)

```
1
11 #ifndef DARWIN_PROJECT_POPULATION_H
12 #define DARWIN_PROJECT_POPULATION_H
13
14 #include "individual.h"
15
21 class Population{
22
23     vector<Individual> population;
27 public:
28
32     Population();
38     bool read_population_from_file(const string& filename);
39
40
45     void crossingover(int k);
46
53     void fitnesscheckpopulation(double extinction_threshold, double proliferation_threshold);
54
60     void writeToFile(const std::string& OutputFile);
61
68     static int randomInteger(int begin, int end);
69
70
71 };
72
73 #endif //DARWIN_PROJECT_POPULATION_H
```





# Index

- chromosomes
  - Individual, [7](#)
- crossingover
  - Population, [9](#)
- fitness\_check
  - Individual, [6](#)
- fitnesscheckpopulation
  - Population, [9](#)
- Individual, [5](#)
  - chromosomes, [7](#)
  - fitness\_check, [6](#)
  - Individual, [6](#)
  - operator<<, [7](#)
- Individual.cpp, [13](#)
  - operator<<, [14](#)
- Individual.h, [14](#)
- main.cpp, [16](#)
- operator<<
  - Individual, [7](#)
  - Individual.cpp, [14](#)
- Population, [8](#)
  - crossingover, [9](#)
  - fitnesscheckpopulation, [9](#)
  - Population, [9](#)
  - population, [11](#)
  - randomInteger, [10](#)
  - read\_population\_from\_file, [10](#)
  - writeToFile, [10](#)
- population
  - Population, [11](#)
- Population.cpp, [17](#)
- Population.h, [18](#)
- randomInteger
  - Population, [10](#)
- read\_population\_from\_file
  - Population, [10](#)
- writeToFile
  - Population, [10](#)