In the question, your task is to complete the following code to make sure that it can be compiled and output the right answer.

#include <iostream>

using namespace std;

// A macro to disallow the copy constructor and operator= functions

// This should be used in the private: declarations for a class

#define DISALLOW\_COPY\_AND\_ASSIGN(TypeName) \

  TypeName(const TypeName&);               \

  void operator=(const TypeName&)

class Doubleton {

 public:

  // Return the instances[0] and instances[1] alternately. For example, the

  // first time calling will return instances[0], the second will return

  // instances[1], the third will return instances[0]... and so on.

  static Doubleton\* getInstance();

  void foo();

 private:

  friend class AtExit;

  static Doubleton\* instances[2];

  static int counter;

  static void deleteInstances();

  Doubleton(int id);

  DISALLOW\_COPY\_AND\_ASSIGN(Doubleton);

  int id\_;

};

void Doubleton::foo() {

  cout << id\_ << endl;

}

// Your submitted code will be pasted here.

class AtExit {

 public:

  AtExit() {}

  ~AtExit() {

    Doubleton::deleteInstances();

  }

};

int main(int argc, char \*argv[]) {

  AtExit a;

  for (int i = 0; i < 10; ++i) {

    Doubleton::getInstance()->foo();

  }

  return 0;

}

Sample Output

|  |
| --- |
| 0  1  0  1  0  1  0  1  0  1 |

You need to complete the following items and submit it.

1. Constructor of class Doubleton;

2. Static member variables

3. Static member functions.

Hint: static member variables and member functions; new and delete; Copy Control.