

Being given the next class definition

```
#include <iostream>
using namespace std;

class Person {
    char* name;
    int age;
    const int id;
    static int noPersons;

public:
    Person(char* Name, int Age, int Id):id(Id) {
        this->name = new char[strlen(Name) + 1];
        strcpy(this->name, Name);
        this->age = Age;

        Person::noPersons++;
    }

    void display() {
        cout << endl << this->name << "(" << this->id << ") has an age of "
             << this->age;
    }

    void setAge(int value) {
        if (value > 5)
            this->age = value;
        else
            throw new exception("Wrong value for age");
    }

    void setName(char* newName) {
        delete [] this->name;
        this->name = new char[strlen(newName) + 1];
        strcpy(this->name, newName);
    }

    static int getNoPersons() {
        return Person::noPersons;
    }

    ~Person() {
        Person::noPersons--;
    }
};

int Person::noPersons = 0;
```

1. Test following statements

	Expected value/error	Actual result/error	Reason for heaving an error
Person p1;			
Person p2("John", 22, 1); p2.display();			
Person *pp1 = new Person("Alice", 23, 2); pp1->display();			
Person p3 = p2; p3.display();			
p3.setAge(5); p3.display();			
p3.setAge(99); p3.display();			
p2.display();			
p3.setName("Mike"); p3.display();			
p2.display();			
Person p4("No name", 0, 0); p4 = p3; p4.setAge(99); p4.setName("Donald");			
p4.display();			
p3.display();			
Person *pp2 = pp1; pp2->display();			
pp2->setAge(25); pp2->setName("Kate");			
pp1->display();			
pp2->display();			

2. Implement the class Copy Constructor and overload = and then retest the same set of instructions. What is the difference?

The copy constructor is

The = operator is

3. Test in main() the next sequence and watch your applications resources in Task manager - > Processes tab.

```
for (;;) {  
    Person* testPerson = new Person("Vader", 21, 1);  
    delete testPerson;  
}
```

Is your application running forever? Why or why not?

3. Correct the class destructor and re-run the previous test. Is anything different?

The correct form of the destructor is:

4. Test following statements with object arrays. From previous exercises keep the copy constructor, = operator and the destructor (the correct form)

	Expected value/error	Actual result/error	Reason for heaving an error
Person p2("John", 22, 1); Person *pp1 = new Person("Alice", 23, 2);			
Person list1[10];			
Person *list2 = new Person[5];			
Person *list3 = new Person("John Doe", 0, 0)[5];			
Person *list4 = new Person[2]{Person("Bob", 22, 3), Person("Vader", 23, 4) };			
Person **list5 = new Person*[3];			
cout << endl << "No of persons:" << Person::getNoPersons();			
list5[0] = &p2; list5[1] = pp1; cout << endl << "No of persons:" << Person::getNoPersons();			
list5[2] = new Person("Bianca", 21, 6); cout << endl << "No of persons:" << Person::getNoPersons();			
delete [] list4; delete [] list5; cout << endl << "No of persons:" << Person::getNoPersons();			
cout << endl << "What is the size (total number of Bytest) of the memory leak (if any)?";			