Subject 3

Observations:

- Maximum working time: 1:15 hours (1 hour and 15 minutes)
- The given code sequences must be seen as pseudo-code (copy pasting them in your project might generate compiler errors)
- All the requirements must be tested in main to be considered for evaluation (if the method is not tested in main it will not be evaluated)

1 pt. Create a C++ project in Microsoft Visual Studio environment that does not generate errors at compile time.

1 pt. Implement the class **FlowerShop** with the next attributes: **Class attributes are defined in the class private space. Set proper default values to the class attributes**

- *shop Id* – integer CONSTANT;
- *name* **– dynamic char array** representing the flower shop name;
- *type* – value from the LOCATION, ONLINE enumeration
- *hasWebsite* – boolean used to indicate if the shop has a website; by default, is false
- *url* – string representing the shop online address;
- *number of sold flowers* – integer that represent the total number of flowers (all the types) sold by the shop.
- *AVERAGE_FLOWER_PRICE* – **static and constant** attribute with the value 7.5

0.5 pts. Implement in class **FlowerShop** ONLY the argument-based constructor that allows the next line. The constructor validates the input.

```
FlowerShop f1(1,"Maria", ONLINE, 15);
// 1 is the shop id
// "Maria" is the shop name
// ONLINE is the symbol of the enumeration
// 15 is the number of sold flowers
```

0.5 pts. Implement the class destructor that will prevent memory leaks.

0.5 pts. Implement the **copy constructor** and public interfaces for accessing private attributes *name* (read and write), *id* (read) and *number of sold flowers* (read and write).
0.5 pts. Test the copy constructor creating a copy of f1

0.5 pts. Implement the next 2 methods

```
f1.openWebsite("www.maria.ro");                 //method will also set the hasWebsite attribute
float totalRevenue = f1.getTotalRevenue();      //total value of sold flowers
```

0.5 pts. Overload the **= operator** without generating memory leaks and avoiding self-reference

Test it doing this

```
FlowerShop f2(2,"Daisy", LOCATION, 35);
f1 = f2;
```

1 pct. Overload stream operators **<< and >>.** The >> operator is used to read all possible attributes from console. The << operator will print all the data on a single line.

```
cin >> f1;
cout << f1;
```

1 pct. Overload the **+ operator** that will add the received value to the total sold flowers and will allow the next expression.

```
f2 = 23 + f1;
cout << f1;
cout << f2;
```

1 pct. Overload **operator +=,** that will increase the number of sold flowers with the received integer.

```
f1+=47;
cout << f1;
```

1 pct. Overload **++ (post incrementation)** to increment the number of sold flowers.

```
f1++;
cout << f1;
```