Subject 4

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 **Hotel** with the next attributes: **Class attributes are defined in the class private space. Set proper default values to the class attributes**

- *hotel Id* – integer CONSTANT;
- *name* **– dynamic char array** representing the hotel name;
- *type* – value from the STARS_3, STARS_4, STARS_5 enumeration
- *isOnBooking* – boolean used to indicate if the hotel is available on booking websites
- *url* – string representing the hotel online address.
- *number of rooms* – integer that represent the total number of rooms.
- *AVERAGE_ROOM_PRICE* – **static and constant** attribute with the value 150.5

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

```
Hotel h1(1,"Inter", STARS_4, 45, "www.inter.ro");
// 1 is the hotel id
// "Inter" is the hotel name
// STARS_4 is the symbol of the enumeration
// 45 is the number of rooms
// "www.inter.ro" is the hotel URL
```

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 rooms* (read and write).
0.5 pts. Test the copy constructor creating a copy of h1

0.5 pts. Implement the next 2 methods

```
h1.registerOnBooking();                    //method sets the isOnBooking attribute
float totalRevenue = h1.getTotalRevenue(); //total value of available rooms
```

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

Test it doing this

```
Hotel h2(2,"Daisy", LOCATION, 35);
h1 = h2;
```

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 >> h1;
cout << h1;
```

1 pct. Overload the *** operator** that will multiply the rooms number and will allow the next expression.

```
h2 = 2 * h1;
cout << h1;
cout << h2;
```

1 pct. Overload **operator -=,** that will decrease the number of rooms with the received integer.

```
h1 -= 10;
cout << h1;
```

1 pct. Overload **++ (pre incrementation)** to increment the room number.

```
++h1;
cout << h1;
```