Name and First name: ____________________________ Group: ___________

Subject 1

*Only solutions that do not contain compile errors will be evaluated.
**Only the methods tested in main() will be evaluated.

Implement the **GeneralStore** class

- ***storeId*** – code represented by a constant integer,
- ***location*** – static array of 40 characters or a string that describes the store location,
- ***brand*** – enumeration with MEGA_IMAGE, KAUFLAND, LIDLE, PENNY types
- ***isHypermarket*** – boolean that marks if the store is a hypermarket, by default is not
- ***receiptsValue*** – variable vector, dynamically allocated, with float numbers representing the value of each client receipt,
- **noClients** – integer that records number of clients
- ***RECEIPT_MAX_VALUE*** – **constant static** real number representing the maximum value of a receipt, initialized with 15000

and develop the C ++ application that contains the following instructions in main():

1pt	Class definition and attributes as required. -0.5 pts of enumeration is not defined and used
1pt	Implement destructor to avoid memory leaks

1pt	**void main() {**
	GeneralStore store1(10, “Bucharest”, LIDLE); //10 is the storeId
	cout << store1.getLocation(); // prints Bucharest
	int receipts[]{ 201, 356.8, 400.5, 100};
	GeneralStore store2(20, “Ploiesti”, receipts, 4, PENNY);
	cout << "\n"<<store2.getReceiptsValue(); //prints 1058.3

1pt	GeneralStore store3 = store2;
	cout << "\n" << store3.getId(); //prints 20
	store3.ExpandToHypermarket() //makes the store a hypermarket

1pt	store3.registerReceipt(300); //add a new client receipt with the value 300
	//validate the receipt value not to exceed
	// RECEIPT_MAX_VALUE
	cout << "\n" << store3.getNoClients(); //prints 5

1pt	store1 = store3;
	cout << "\n" << store1.getReceiptValue(2); //prints the value of the n-th
	// receipt

1pt	cout << store1; //prints store data
	// - the brand must be printed as a string
	// - if isHypermarket print a message

1pt	float averageValue = (float)store1; //computes the average value of receipts
	cout << "\n" << averageLiters;

1pt	store1 -= 10; //updates all receipts value
	//lowering it with 10, but not below 0
	cout << store1;

1pt	if (store1 > 5000)
	       cout << "\n Store 1 has sold more than 5000";
	else
	       cout << "\n Store 1 has sold less than 5000";