COMP 322
Winter Semester 2019

INSTRUCTOR: DR. CHAD ZAMMAR
chad.zammar@mcgill.ca

Assignment 2: Exploring Classes.

Due date: 11 March 2019, 11:59 PM.

Before you start:

- Collaboration and research for similar problems on the internet are recommended. However, your submission should reflect individual work and personal effort.
- Some of the topics may not be covered in class due to our limited time. You are encouraged to find answers online. You can also reach to your instructor or TAs for guidance.
- Please do submit your assignment before the due date to avoid penalties or worse risking your assignment being rejected.
- Submit only one .cpp file to make it easier to grade. If some questions require you to develop and explain, you can embed your text in a C++ comment style.
- Make sure your code is clear and readable. **Readability of your code as well as the quality of your comments will be graded**.
- No submission by email. Submit your work to mycourse.
- If your code does not compile it will not be graded.
- Be happy when working on your assignment, because a happy software developer has more inspiration than a sad one :).

In the following questions you will be designing a class called **SuperDraw** to generate lotto numbers and manage the process of verifying whether a ticket is a winner. Every lotto ticket has 6 numbers generated randomly between 1 and 49. A number cannot be repeated in the same ticket.

The main SuperDraw class structure should look like the following:

```cpp
struct ticket
{
       unsigned int numbers[6];
       ticket* next;
};

class SuperDraw
{
public:
       SuperDraw();
       ~SuperDraw();
private:
       ticket* ticketListHead;
       ticket* ticketListTail;
};
```

ticket is a linked list structure that holds the 6 lotto numbers in an array and a pointer to the next element in the list. Notice that the class SuperDraw has 2 private data members, ticketListHead and ticketListTail. They are 2 pointers pointing to the head (first element) and the tail (last element) of the linked list.

Question 1 (5 pts)

Complete the implementation for the class **SuperDraw** by implementing the constructor and the destructor bodies if needed. Constructor should be initializing the object to whatever initial suitable state.

Question 2 (15 pts)

Add a public method called newTicket(int verbose = 0) that generates random 6 numbers. The newly created ticket should be added to the linked list and the randomly generated numbers should be printed out to the screen if verbose argument is set to 1. By default the verbose argument is set to 0 which means that no messages will be printed out to the screen.

The numbers should be sorted in ascending order.

Remember that the pointers ticketListHead and ticketListTail should be updated accordingly after the generation of each new ticket

The test main() function should look like the following:

```cpp
int main()
{
    SuperDraw sd;
    sd.newTicket(1);
}
```

The output should be something like:

A new ticket was successfully generated. The numbers are: 12, 14, 23, 39, 40, 44

Question 3 (10 pts)

Add a constructor that takes an int argument which corresponds to the number of tickets to be generated.

The test main() function should look like the following:

```cpp
int main()
{
    SuperDraw sd(2);
}
```

The output should be something like:

2 new ticket were successfully generated.

The numbers are: 12, 14, 23, 39, 40, 44 and 1, 2, 9, 12, 28, 41

Question 4 (15 pts)

Add a public method called printAllTicketNumbers() that print to the screen a list of all generated numbers for all the tickets.

The test main() function should look like the following:

```cpp
int main()
{
    SuperDraw sd;
    sd.newTicket();
    sd.newTicket();
    sd.newTicket();
    sd.newTicket();
    sd.printAllTicketNumbers();
}
```

The output should be something like:

We found 4 generated tickets:

12, 14, 23, 39, 40, 44

1, 5, 12, 14, 32, 33

2,24, 27, 29, 45, 46

8, 12, 19, 29, 32, 34

Question 5 (15 pts)

Add a method called verifySequence() that verifies if a certain sequence of numbers is already generated.

The test main() function should look like the following:

```
int main()
{
    SuperDraw sd;
    sd.newTicket();
    // as many sd.newTicket() as you like
    Int myNumbers[6] = {2, 4, 17, 29, 31, 34}
    sd.verifySequence(myNumbers)
}
```

The output should be something like:

The provided sequence of numbers was never generated before

Or:

The provided sequence of numbers was already generated.

Question 6 (15 pts)

Add a method called deleteSequence() that deletes a ticket. Note that in order to delete a ticket, you have to verify first whether a ticket existed, if so you need to delete it by freeing its allocated memory and then you need to update the *next* pointer of the previous element in the list that was initially pointing to it.

The test main() function should look like the following:

```
int main()
{
    SuperDraw sd;
    sd.newTicket();
    // as many sd.newTicket() as you like
    Int myNumbers[6] = {2, 4, 17, 29, 31, 34}
```

```
        sd.deleteSequence(myNumbers)
}
```

The output should be something like:

The provided sequence of numbers was never generated before

Or:

The provided sequence of numbers was successfully deleted.

Question 7 (10 pts)

Provide an implementation for the destructor method that ensures that all previously allocated tickets are freed when the object is destroyed so that we don't risk having memory leaks after running the program.

Question 8 (15 pts)

Provide a copy constructor that copies the content of a SuperDraw object into another object.

The test main() function should look like the following:

```cpp
int main()
{
    SuperDraw sd;
    sd.newTicket();
    // as many sd.newTicket() as you like
    SuperDraw sd2(sd);
    sd2.printAllTicketNumbers();
}
```