CENG 241 Lab 7

Q1-)

Book
-title: string
-writer: string
-price: float
+Book(): Default Const
+Book(string, string, float): Overloaded Const
+Book(Book &obj): Copy Const

Write a C++ program that has a Book class which is shown in the above figure. Class has default constructor, as well as overloaded constructor. There are three variables in the class title, writer and price. Moreover, There is a copy constructor in class. You are also required to write getters and setters for all class variables.

In your main function you need to create an object of array with the size of 2. And fill the array with the user inputs. After that, let user to copy first element or second element of the array. Then, print the book info respectively. As last, print the copied book to the screen.

Sample Run:

Enter Title: ***Millions and Billions***

Enter Writer: ***Brian COX***

Enter Price: ***120***

Enter Title: ***Art Of War***

Enter Writer: ***Sun TZU***

Enter Price: ***65***

The book you wanna copy: 1 or 2

1

book 1:

Millions and Billions Brian COX 120

book 2:

Art Of War Sun TZU 65

copy book :

Millions and Billions Brian COX 120

CENG 241 Lab 7

Q2-)

SpecialDay

-year: int

-month: int

-day: int

-title: string

-repetitive: bool

+SpecialDay(): Default Const

+SpecialDay(string,int,int,int): Overloaded Const

+SpecialDay(SpecialDay &obj): Copy Const

Write a C++ program to store special days inputted by the user. If the day is a repetitive day it needs to be saved for another year and print to the screen. You must have a SpecialDay class and need to have all getter and setters for class variables.You need to have default and overloaded constructor as well. You need a dynamic array of objects in your main class and determine the size of the array in runtime by the user input. Beside this, you are required to have a copy constructor in your class and for repetitive days you need to copy the objects and increase their value of year by one and print them.

Sample Output: (Bold texts are inputs entered by the user)

Enter number of days to record: ***3***

Enter title, day, month and year respectively:

New Year

01 01 2022

Is the day repetitive?

yes

Enter title, day, month and year respectively:

Black Friday

26 11 2022

Is the day repetitive?

yes

Enter title, day, month and year respectively:

Lab7 Due Date

29 11 2021

Is the day repetitive?

No

Your Special Occasions:

New Year 1/1/2022

Black Friday 26/11/2022

Lab7 Due Date 29/11/2021

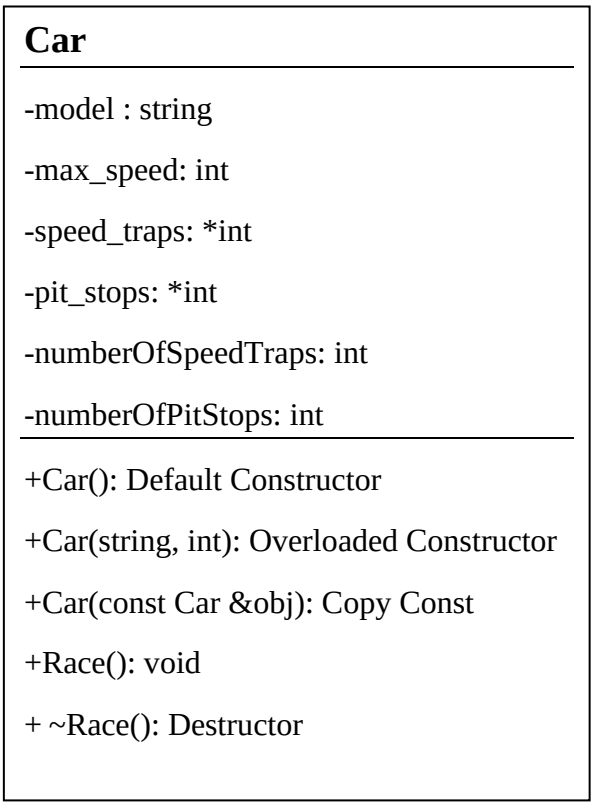
Repetitive Occasions:

New Year 1/1/2023

Black Friday 26/11/2023

CENG 241 Lab 7

Q3-)



Write a C++ program that has a Car class shown above figure. There are attributes of the car like model, max_speed and there are **5 speed traps** to record the speed of the car in certain part of the circuit that hosts race cars. Each car needs to do **2 pit stops** to refuel and changing their tyres. You need to write getter and setters for each variables of the class. In default constructor, you need to give initial values for max speed as "100" and model as empty. However in overloaded constructor, model and max_sped variables need to be taken with parameters to change values of class variables. Both speed traps and pit stops need to store in dynamic arrays inside Car class. You need to have a copy constructor and use this copy constructor in main method to copy your Car object to another (You need to use **deep copy** for dynamic arrays.) and you need to call Race() function for the object that copied from the first object. In Race() method, program should fill numberOfSpeedTraps array by random values between 10 and max speed of the car. Also, time to make pitstops must be determined by random values between 1 and 10. Car's speed is in kilometers. Pit stops' time are calculated in seconds. Assume race circuit has 50 km long. As last, you need to find average speed of the cars and find the time that each car finishes the circuit in how many minutes and seconds. Then, find the winning car and print to the screen. In Destructor mehod you need to delete dynamic arrays from the memory.

Sample Output:

```
Car 1:
speed trap:
1: 86 pit stop: 9
2: 62 pit stop: 8
3: 51 4: 10 5: 93 Avg Speed: 60.4
Total time: 49: 57
Car 2:
speed trap:
1: 5 pit stop: 8
2: 58 pit stop: 10
3: 99 4: 100 5: 41 Avg Speed: 60.6
Total time: 50: 22
Winner is Car 1
```