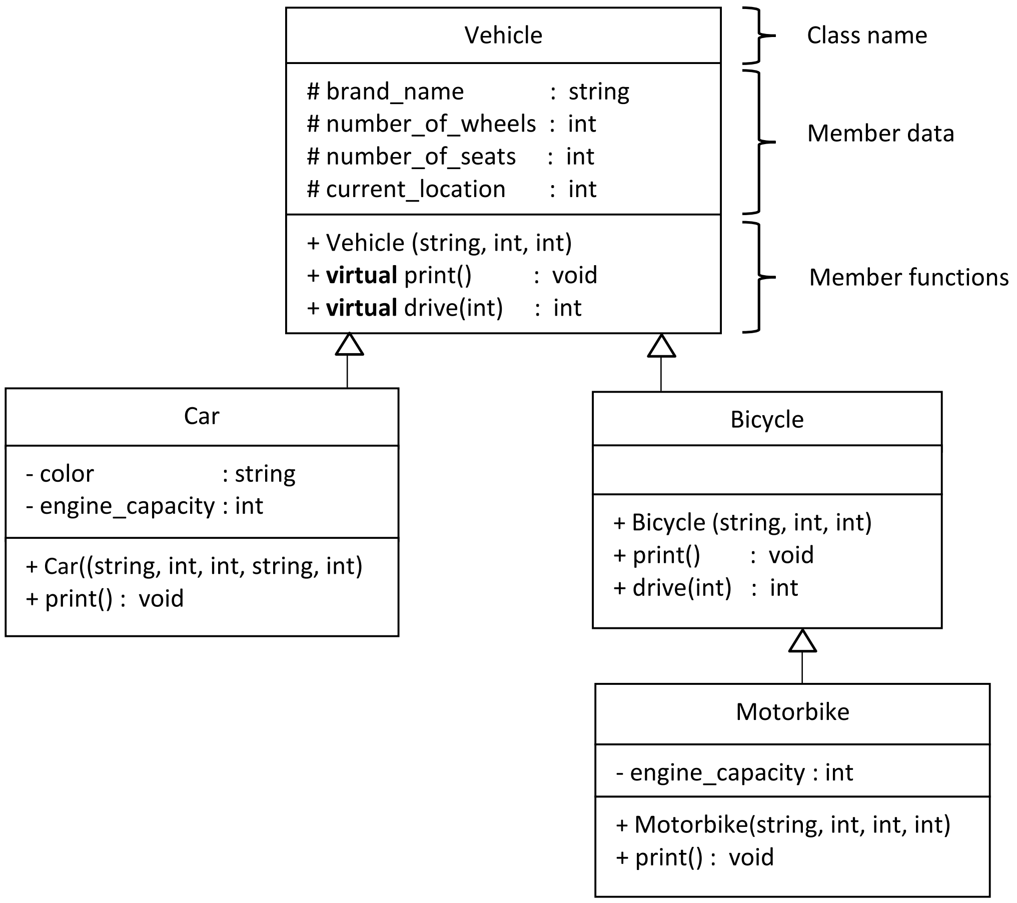
BLG252E - Object Oriented Programming

Homework-2

Assignment Date :	29.11.2016
Due Date :	20.12.2016 at 18:00

The following UML class diagrams show the class names, data members, member functions, and the hierarchy of inheritances. (Member access symbols : - is private, + is public, # is protected).
All derivations are as public inheritance.



Write C++ codes for the classes with the following specifications.

- Parameterized constructor prototypes:

  Vehicle (string brand_name, int number_of_wheels, int number_of_seats);
  Car (string brand_name, int number_of_wheels, int number_of_seats, string color, int engine_capacity);
  Bicycle (string brand_name, int number_of_wheels, int number_of_seats);
  Motorbike (string brand_name, int number_of_wheels, int number_of_seats, int engine_capacity);

- Car and Bicycle class constructors should display warning messages when invalid parameters have been passed, but should accept them anyway.
  - A Car object must have at least 4 seats and 4 wheels.
  - A Bicycle/Motorbike object must have exactly 2 wheels, also 1 or 2 seats.

- The current_location member should be initialized to zero by Vehicle class constructor.

- When a vehicle is driven, the current location is changed. The drive function takes a distance as parameter and advances the current location by the distance and returns the updated current location.

    - Car can drive forward (+) and backward (-) .
    - Bicycle and Motorbike can drive only forward (+) .
    - You should override the drive function for Bicycle, so that an error message is displayed when a negative distance is passed as parameter. In that case, the current location should not be updated.

- The print functions of derived classes should display the type of the vehicle (car, bicycle, or motorbike). Also the print function of top base class (Vehicle) should be called.

MAIN PROGRAM

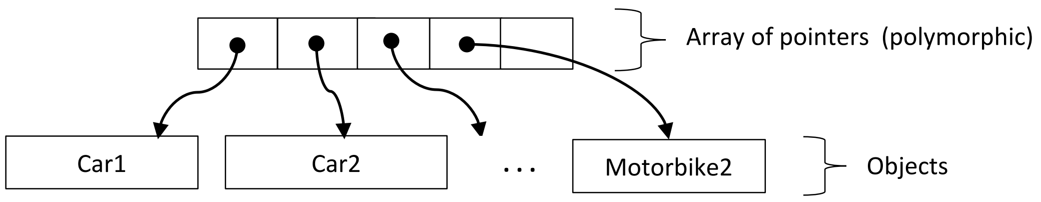
Write the main C++ program to do followings.

- **PHASE1:** Define an array of Vehicle pointers. (This polymorphic array will be used to store pointers to different types of vehicles.) With dynamic allocation, construct the following objects and store their memory addresses in the polymorphic array.

- **PHASE2:** For each object in the polymorhic array; first call the drive function with 200 km distance, then call the drive function again with -50 km distance.

- **PHASE3:** By looping, call the print function for each object in the polymorhic array.

Vehicle type	Brand name	Number of wheels	Number of seats	Color	Engine capacity
Car	"Car1"	4	4	"Color1"	1400
Car	"Car2"	4	5	"Color2"	1600
Car	"Car3"	4	4	"Color3"	1200
Car	"Car4"	**3**	4	"Color4"	1300
Car	"Car5"	4	4	"Color5"	1700
Bicycle	"Bicycle1"	2	1		
Bicycle	"Bicycle2"	2	10		
Motorbike	"Motorbike1"	1	2		900
Motorbike	"Motorbike2"	2	1		700



EXAMPLE SCREEN OUTPUT

By using the ***setw*** and the ***left*** manipulators (include the <iomanip> header file), generate a screen output which is as similar as possible to the example given below.

```
PHASE1 : CONSTRUCTINGS
Car4            --> Constructor warning : Car can not have 3 wheels!
Bicycle2        --> Constructor warning : Bicycle/Motorbike can not have 10 seats!
Motorbike1      --> Constructor warning : Bicycle/Motorbike can not have 1 wheels!

PHASE2 : DRIVINGS
Bicycle1        --> Drive error : Bicycle/Motorbike can not drive -50 direction!
Bicycle2        --> Drive error : Bicycle/Motorbike can not drive -50 direction!
Motorbike1      --> Drive error : Bicycle/Motorbike can not drive -50 direction!
Motorbike2      --> Drive error : Bicycle/Motorbike can not drive -50 direction!

PHASE3 : PRINTINGS
(Vehicle type , Brand name , Number of wheels , Number of seats , Current location ,
Color , Engine capacity)
Car       : Car1                4  4    150     Color1  1400
Car       : Car2                4  5    150     Color2  1600
Car       : Car3                4  4    150     Color3  1200
Car       : Car4                3  4    150     Color4  1300
Car       : Car5                4  4    150     Color5  1700
Bicycle   : Bicycle1            2  1    200
Bicycle   : Bicycle2            2  10   200
Motorbike : Motorbike1          1  2    200             900
Motorbike : Motorbike2          2  1    200             700

Press any key to continue . . .
```

IMPORTANT RULES ABOUT BLG252E HOMEWORKS

1) You must do the homeworks by yourself individually.

- Copying, collaboration, getting help is absolutely not permitted.
- A student should never give his homework to other students.
- All submitted student homework files will be compared by using an automatic detection software system (such as Moss, JPlag, etc).
- If significant amount of similarities are found between any files, it will be considered as cheating; and those homework grades will be zero.

2) You should submit your homework file to Ninova system only.

- Email submissions or late submission requests are not accepted.
- Ninova homework system closes itself automatically at the deadline time.
  Therefore you should not wait for homework submission until the last minutes.
- You should submit only a file with *.cpp extension to Ninova.
  Other types of files (such as c, txt, docx, zip, rar, etc.) are not accepted.
- If you make any changes in your homework file, you can re-submit it to Ninova within the deadline time.
  In that case, only the last submitted file is kept in the system by Ninova.

3) Homeworks will be graded by the course assistant and results will be announced at Ninova.

4) The following criteria will be considered when grading the homeworks.

Your program should ;

- be compilable with all standard compilers (Dev-C++, Linux, etc.) without any syntax errors.
- not include non-portable header files such as <conio.h> , <stdafx.h> , etc.
- work correctly, effectively, and display expected outputs.
- be written according to given specifications.
- have a consistent coding style (indentation, comment lines, valid variable names).
- contain the following information at the beginning of your source file.
  (otherwise 5 points will deducted from the homework grade).

```
/***************************
Student Number : 123456789
Student Name   : Aaa Bbb
Course         : BLG252E
CRN            : 12345
Term           : 2016-Fall
Homework       : #2
***************************/
```