CS1004 - Object-oriented Programming (OOP)

Assignment # 3

Max Points: **60**

Due Date: **Sunday, December 4, 2021, 11:59 p.m.**

**Carefully read the following instructions!**

* It should be clear that your assignment would **not get any credit** if the assignment is submitted after the **due date**. **No** assignment will be **accepted after the due date**.
* Strict action will be taken if the submitted solution is copied from any other student.
* If you people find any mistake or confusion in the assignment (Question statement), please consult before the deadline. After the deadline no queries will be entertained in this regard.
* For any query, feel free to email at: **farah.sadia@nu.edu.pk**
* Submission: Submission will only be accepted through **GOOGLE CLASSROOM**.
* Submit all your codes with your Student ID and task number. **“K211234\_Q1”**.
* Every code should be with proper **commenting**.

**Question # 01:(Short Question Answer)**

1. Can we declare function parameters as static?
2. Can we make a destructor virtual? What are the benefits of it?
3. Can we make a virtual constructor?
4. What is name mangling?
5. Define late binding & early binding.
6. Default parameters cause problems when there are overloaded functions. How?
7. Can we override a constructor?
8. When does a destructor get called?
9. Can we call a destructor explicitly?
10. Is there any difference between a null pointer & a void pointer?
11. Do friend functions violate the principle of encapsulation?
12. What’s the value of i++ + i++?
13. Can we stop a class from getting inherited?
14. Can we call a virtual function from a constructor?

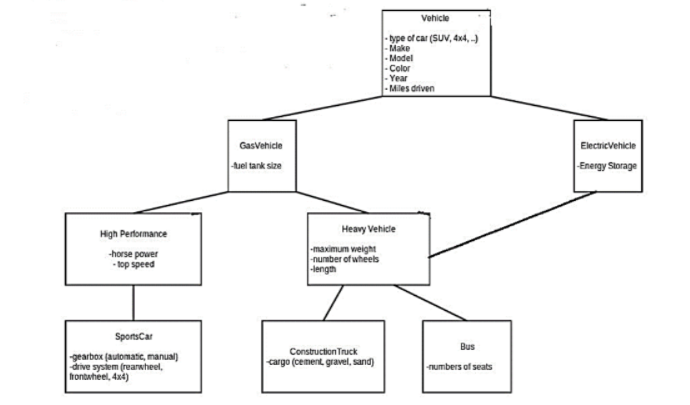
**Question # 02:(Method Overloading)**

Create a Message class with a constructor that takes a single string with a default value. Create a private member string, and in the constructor simply assign the argument string to your internal string. Create two overloaded member functions called print( ): one that takes no arguments and simply prints the message stored in the object, and one that takes a string argument, which it prints in addition to the internal message.

**Question # 03:(Method Overriding)**

All the values are required to be set through the constructor's parameter.

1. Provide necessary accessor functions where required.
2. Create an object of class bus by initializing it through a parameterized constructor in the main function and display.
3. All data members by calling the display function of the class bus.



**Question # 04:(Operator Overloading)**

Design a class to perform various matrix operations. A matrix is a set of numbers arranged in rows and columns. Therefore, every element of a matrix has a row position and a column position. If A is a matrix of five rows and six columns, we say that the matrix A is of the size 5x6 and sometimes denote it as A 5x6.

Clearly, a convenient place to store a matrix is in a two-dimensional array. Two matrices can be added and subtracted if they have the same size. Suppose **A = [a**ij**] and B = [b**ij**]** are two matrices of the size in x n, in which a ij denotes the element of A in the i th row and the j th column and so on. The sum and difference of A and B are given by:

**A+B = [a**ij **+ b**ij **]**

**A-B = [a**ij **– b**ij **]**

The multiplication of A and B (A\*B) is defined only if the number of columns of A is the same as the number of rows of B. If A is of the size m \* n and B is of the size n \* t, then **A \* B = [c**ik **]** is of the size m\*t and the element c ik is given by the formula:

**C**ik **= a**i1 **b** 1k **+ a**i2 **b**2k **+ … + c**in **b**nk

Design and implement a Class matrixType that can store a matrix of any size. Overload the operators +, - and \* to perform the addition, subtraction, and multiplication operations, respectively, and overload the operator &lt;&lt; to output a matrix. Also, write a test program to test various operations on the matrices.

**Question # 05:(Runtime Polymorphism)**

Write a program to calculate the bonus of the employees. The class master derives the information from both admin and account classes which intern derives information from the class person. Create base and all derived classes having the same member functions called getdata, display data and bonus. Create a base class pointer that is capable of accessing data of any class and calculates the bonus of the specified employee. (Hint: Use virtual functions)