Assignment 1:

Vehicle:(Parent)

- create two abstract methods - run() and stop()

- create three methods of public void fuel - 1st method will take int / 2nd method will float and string / 3rd method will take char and int (method overloading)

- declare two variables - int speed and long distance

- create two constructors also- default and parameterized

2W:

- override the two methods - run() and stop()

- create default constructor

- declare two variables - int speed and long distance with different values from the parent vehicle

- declare one more variable as int num\_of\_tire = 2

- create methods here also as display() --> this will print the three variables of 2W + all the variable of the parent Vehicle (hint: super.variable\_name)

3W:

- override the two methods - run() and stop()

- - create default constructor

- declare two variables - int speed and long distance with different values from the parent vehicle

- declare one more variable as int num\_of\_tire = 3

- create methods here also as display() --> this will print the three variables of 3W + all the variable of the parent Vehicle

4W:

- override the two methods - run() and stop()

- create default constructor

- declare two variables - int speed and long distance with different values from the parent vehicle

- declare one more variable as int num\_of\_tire = 4

- create methods here also as display() --> this will print the three variables of 4W + all the variable of the parent Vehicle

8W:

- override the two methods - run() and stop()

- create default constructor

- declare two variables - int speed and long distance with different values from the parent vehicle

- declare one more variable as int num\_of\_tire = 8

- create methods here also as display() --> this will print the three variables of 8W + all the variable of the parent Vehicle

Main()

- call all the methods using dynamic/runtime polymorphism here

- all the methods of all the child classes.

- call all the methods of the fuel of Vehicle class

=================================================================================================

Assignment 2:

(Parent) MNC ---> 2 abstract methods + 1 constructor + 1 normal method

Infosys (child of MNC) ----> make it abstract also but give the implementation of single abstract method

Hello (child of Infosys) ----> here implement all the abstract methods + 1 create normal method also

main class ---> call all the methods of all classes using the dynamic polymorphism

================================================================================================

Assignment 3:

1) Create an abstract class pen with methods write () and refill () as abstract methods

2) Use the pen class from Q1 to create a concrete class fountain pen with additional method change Nib ()

3) Create a class monkey with jump ( ) and bite ( ) methods Create a class human which inherits this monkey class and implements basic animal interface with eat ( ) and sleep methods

4) Demonstrate polymorphism using using monkey class from Q3

=================================================================================================

Assignment 4:

Create a class telephone with ( ) , lift ( ) and disconnected ( ) methods as abstract methods create another class smart telephone and demonstrate polymorphism

=================================================================================================

Assignment 5:

Create an interface TVremote and use it to inherit another interface smart TVremote

Create a class TV which implements TVremote interface from Q6