NATIONAL UNIVERSITY
OF COMPUTER & EMERGING SCIENCES
LAHORE

Course Name & Section OOP Date 26-3-24

Student's Name Rayan Ahmad KHAN Roll No. 23L-3030 Section BCS-2D

Group - B (see next page)

Q / Part No. | Rough Work

(Turn Over)

15/20

```
class Car {

  char ~~(20)~~ * name;
  int *top_speed;
  bool *electric_car;
  static int cars ;

public:

   Car( name1, top_speed1, e , ~~carno~~)
   {
       name    = name1;
       top_speed   =  top_speed1 ;
       electric_car  =  e;
       cars++ ;
   }
```

(∵ Please turn Over)

```
class  Car {     2+2+0.5+1+2+3+1+1.5+2

char * name;
int  top_speed;
bool  electric_car;
static  int  cars;

public:

 Car (*n , t_s , e,) {
  $ int  l =  strlen(n)
  name  =  new char [l];
  for (int i=0 ; i<l ; i++){
    name [i]  =  n[i] ;
  }

   top_speed  =  t_s ;
  ~~electric~~
   if (e)
    electric_car  =  true;
   else
    electric_car  =  false;
    cars ++ ;
 }

~Car (  )
 {
   delete []  name;
 }      cars--;

  Car( const  Car& rhs) ;
  Car  operator  = (const Car& rhs);
  Car  operator  + (const  Car& rhs);
  void  operator !  ( );
 ~~&~~void  UpgradeCar( const~~&~~ Car&( rhs);
  void  Printinfo( )  ;
 };

Car:: static  int  cars =  nullptr;
                      X  =0
Car:: Cor( const & Cor& rhs)
{
        this -> top_speed = rhs. top-speed;
        for (int i=0 ;name[i] != '\0' ; i++)   × need
          this -> name[i] = rhs. name[i];      shallow
                  e = rhs.e;                   copy.
}.

Car :: Car operator =(const car& rhs){
       Car temp;

      temp. top_speed = (this-> top_speed +   X
       for(int i=0; name[i]!='\0'; i++)  rhs. top_speed);
        temp. name[i] = rhs. name[i];
        return temp;               e = rhs.e;

}

Car :: void operator! () {

      if ( this -> electric car)
          return true;
      return false;

}

Car :: void UpgradeCar(const Car& rhs)
{
       if ( rhs. top_speed > this->top_speed)

       this -> top_speed += rhs. top_speed ;

       else
            error?
            return ;

}
 Car :: Car operator +( const car& rhs)
   Car temp();
   temp. top_speed = this -> top_speed +
                        rhs. top_ speed ;
 } return   temp;
```

```
Car :: void  printInfo () {

 cout << "This  is  a" << this->name //part of
          << " its  top  speed  is " <<   " code is
See below this -> top speed  << "km/h." << ".";  " mentioned
                                                 " below
 if ( this -> electric_car ) {

 cout << "It  is  a  electric car \n." ;

 }

 else
 {
 cout << "It  is  not  an  electric
              car." ;

 cout << endl;

 }
```

```
// remaining  part of code
  for ( int i = 0 ; ~~arr~~ name[i] != '\0' ; i++)
       cout << name[i] ;
```