1) *Create* ***Car*** *class with the following data members:*

`char name[20]`
Car's brand name. *(20 characters default to empty)*

`bool autoTrans`
Transmission type. *(True for automatic transmission. default to false)*

`float liter100Km`
Fuel Consumption. (*Default to 0.0*)

`int weight`
Car weight. (*Default to 0*)

and the following member functions:

`bool isAuto();`
Function that returns the value of data member "**`autoTrans`**"

`float getLper100Km();`
Function that returns the value of data member "**`liter100Km`**"

`void setCar(char[],bool,float,int);`
Function that sets the data members.
1st argument is the name of the car.
2nd argument is the automatic transmission (true/false).
3rd argument fuel consumption.
4th argument weight of the car.
This function should include the following integrity check:
1. "**`name`**" is less than 19 characters.
2. "**`autoTrans`**" is only true/false.
3. "**`liter100Km`**" is a positive number.
4. "**`weight`**" is a positive number.

`void printCar();`
Function that prints the value data members in the following format.

```
Brand name: MGB
--------------------
Transmission: Manual
Liters / 100 Km: 6.3
Wieght: 1000
```

---

2) *Create class* ***Rectangle*** *.The class has attributes* ***length*** *and* ***width*** *, each of which defaults to 1.0. It has member functions that calculate the* ***perimeter*** *and the* ***area*** *of the rectangle. It has a* ***set*** *and* ***get*** *function for both data members. It should also verify that the values of both data members is larger than 0.0 and less than 20.0.*

1)

```
#include <iostream.h>
#include <cstring>

class Car {
  private:
    char name[20];
    bool autoTrans;
    float liter100Km;
    int weight;
  public:
    Car();
    bool isAuto();
    float getLper100Km();
    void setCar(char [], bool, float, int);
    void printCar();
};
```

```
Car::Car() {
    name[0]='\0';
    autoTrans = false;
    liter100Km=0.0;
    weight=0;
}

bool Car::isAuto() {
    return autoTrans;
}

float Car::getLper100Km() {
    return liter100Km;
}

void Car::setCar(char cNam[], bool aTrans,
                          float l100km, int wgt) {
    strncpy(name,cNam,19);
    name[19]='\0';

    autoTrans=  aTrans ? true : false;
    liter100Km = l100km >=0 ? l100km : 0;
    weight = wgt >=0 ? wgt : 0;
}
```

```
void Car::printCar() {
     cout << "Brand name: " << name << endl;
     cout << "----------" << endl;
     cout << "Transmission: "
               << (autoTrans ? "Automatic " : "Manual ")
                << endl;
     cout << "Liters / 100 Km: " << liter100Km
               << endl;
     cout << "Weight: " << weight
               << endl << endl;
}
```

```
void main() {
     Car MyCar, fleet[3], *thisCar;

     MyCar.printCar();

     MyCar.setCar("MGB",false,6.3,1000);
     MyCar.printCar();

     fleet[0].setCar("Honda",false,5.5,1534);
     fleet[1].setCar("Peugeot",true,7.3,2200);
     fleet[2].setCar("Fiat",false,10.3,1322);

     thisCar = &fleet[2];
     thisCar->printCar();

     for (int ndx=0; ndx < 3; ndx++){
          if (fleet[ndx].isAuto()) {
               fleet[ndx].printCar();
          }
     }
}
```

2)

Rect.h

```
#ifndef RECT_H
#define RECT_H

class Rectangle {
  public:
    Rectangle( double = 1.0, double = 1.0 );
    double perimeter( void );
    double area( void );
    void setWidth( double w );
    void setLength( double l );
    double getWidth( void );
    double getLength( void );
  private:
    double length;
    double width;
};

#endif
```

Rect.cpp

```
#include "Rect.h"

Rectangle::Rectangle( double w, double l ) {
   setWidth(w);
   setLength(l);
}

double Rectangle::perimeter( void )  {
  return 2 * ( width + length );
}

double Rectangle::area( void )  {
  return width * length;
}

double Rectangle::getWidth( void ) { return width; }

double Rectangle::getLength( void ) { return length; }
```

```
void Rectangle::setWidth( double w )  {
  width = w > 0 && w < 20.0 ? w : 1.0;
}

void Rectangle::setLength( double l ) {
  length = l > 0 && l < 20.0 ? l : 1.0;
}
```

Prog.cpp

```
#include <iostream.h>
#include <iomanip.h>

#include "Rect.h"

int main() {
   Rectangle a, b( 4.0, 5.0 ), c( 67.0, 888.0 );

   cout << setiosflags( ios::fixed | ios::showpoint );
   cout << setprecision( 1 );

   // output Rectangle a
   cout << "a: length = " << a.getLength()
        << "; width = " << a.getWidth()
        << "; perimeter = " << a.perimeter() << ";area = "
        << a.area() << '\n';

   // output Rectangle b
   cout << "b: length = " << b.getLength()
        << "; width = " << b.getWidth()
        << "; perimeter = " << b.perimeter() << ";area = "
        << b.area() << '\n';

   // output Rectangle c; bad values attempted
   cout << "c: length = " << c.getLength()
        << "; width = " << c.getWidth()
        << "; perimeter = " << c.perimeter() << "; area = "
        << c.area() << endl;

   return 0;
}
```