**LAB TASK # 7**

**Question # 1**

**INPUT :**

#include <string>

#include <iostream>

#include <conio.h>

#define \_USE\_MATH\_DEFINES

#include<math.h>

using namespace std;

class circle

{

public:

double radius;

public:

circle()

{};

circle(double r) :radius(r)

{};

void setradius(double r)

{

radius = r;

}

double getradius()

{

return radius;

}

double calcarea()

{

return M\_PI \* radius \* radius;

}

double calccircumferrencw()

{

return 2 \* M\_PI \* radius;

}

};

class cylinder :public circle

{

private:

double height;

public:

cylinder()

{};

cylinder(double radius, double h) :circle(radius), height(h)

{};

void setheight(double h)

{

height = h;

}

double getheight()

{

return height;

}

double calcarea()

{

return((2 \* M\_PI \* radius \* height) + (2 \* M\_PI \* radius \* radius));

}

double calcvolume()

{

return M\_PI \* radius \* radius \* height;

}

};

int main()

{

circle\* ptr;

ptr = new circle(3.3);

cout << "The area of circle is\t" << ptr->calcarea() << endl;

cylinder\* ptr1;

ptr1 = new cylinder(4.2, 10.0);

cout << "The area of cylinder is\t" << ptr1->calcarea() << endl;

}

**OUTPUT :**



**Question # 2**

**INPUT :**

#include <string>

#include <iostream>

#include <conio.h>

using namespace std;

class doors

{

private:

int no\_of\_doors;

string door\_lock;

public:

doors()

{};

doors(int nod, string dl) :no\_of\_doors(nod), door\_lock(dl)

{};

void setno\_of\_doors(int nod)

{

no\_of\_doors = nod;

}

int getno\_of\_doors()

{

return no\_of\_doors;

}

void setdoor\_lock(string dl)

{

door\_lock = dl;

}

string getdoor\_lock()

{

return door\_lock;

}

void showdata()

{

cout << “Car has “<<no\_of\_doors<<” doors.”<< endl;

cout <<”Car has “<< door\_lock <<” door lock.”<< endl;

}

};

class windows

{

private:

int no\_of\_windows;

int manual\_windows;

int auto\_windows;

public:

windows()

{};

windows(int now, int mw, int aw) :no\_of\_windows(now), manual\_windows(mw), auto\_windows(aw)

{};

void setno\_of\_windows(int now)

{

no\_of\_windows = now;

}

int getno\_of\_windows()

{

return no\_of\_windows;

}

void setmanual\_windows(int mw)

{

manual\_windows = mw;

}

int getmanual\_windows()

{

return manual\_windows;

}

void setauto\_windows(int aw)

{

auto\_windows = aw;

}

int getauto\_windows()

{

return auto\_windows;

}

void showdata()

{

cout <<”Car has “<< no\_of\_windows<<” windows out of which “<<manual\_windows<<” are manual and “<<auto\_windows<<” are auto windows.” << endl;

}

};

class engine

{

private:

int h\_power;

int oil\_capacity;

public:

engine()

{};

engine(int hp, int oc) :h\_power(hp), oil\_capacity(oc)

{};

void seth\_power(int hp)

{

h\_power = hp;

}

int geth\_power()

{

return h\_power;

}

void setoil\_capacity(int oc)

{

oil\_capacity = oc;

}

int getoil\_capacity()

{

return oil\_capacity;

}

void showdata()

{

cout <<”Car has “ <<h\_power<<”HP engine and it’s oil capacity is “<<oil\_capacity<<”L.” << endl;

}

};

class wheel

{

private:

string wheel\_type;

string rim\_type;

public:

wheel()

{};

wheel(string wt, string rt) :wheel\_type(wt),rim\_type(rt)

{};

void setwheel\_type(string wt)

{

wheel\_type = wt;

}

string getwheel\_type()

{

return wheel\_type;

}

void setrim\_type(string rt)

{

rim\_type = rt;

}

string getrim\_type()

{

return rim\_type;

}

void showdata()

{

cout <<”Wheel type of car is “<< wheel\_type<<” and rims are “<<rim\_type <<”.”<< endl;

}

};

class car

{

private:

doors door;

windows window;

engine eng;

wheel whe;

public:

car()

{};

car(doors d1, windows w1,engine e1,wheel we1)

{

door = d1;

window = w1;

eng = e1;

whe = we1;

}

void showdata()

{

door.showdata();

window.showdata();

eng.showdata();

whe.showdata();

}

};

int main()

{

doors d1(4, “auto”);

windows w1(4, 2, 2);

engine e1(1000,2);

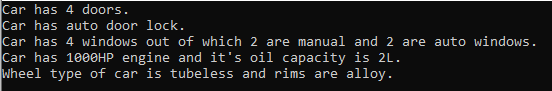
wheel we1(“tubeless”,”alloy”);

car c1(d1, w1,e1,we1);

c1.showdata();

}

**OUTPUT :**



**Question # 3**

**INPUT :**

#include <iostream>

#include <conio.h>

using namespace std;

class box

{

private:

double height;

double length;

double width;

public:

box()

{};

box(double h, double l, double w) :height(h), length(l), width(w)

{};

void setheight(double h)

{

height = h;

}

double getheight()

{

return height;

}

void setlength(double l)

{

length = l;

}

double getlength()

{

return length;

}

void setwidth(double w)

{

width = w;

}

double getwidth()

{

return width;

}

double volume()

{

return length \* width \* height;

}

box operator+(const box& b)

{

box box;

box.height = this->height + b.height;

box.length = this->length + b.length;

box.width = this->width + b.width;

return box;

}

box operator-(const box& b)

{

box box;

box.height = this->height - b.height;

box.length = this->length - b.length;

box.width = this->width - b.width;

return box;

}

box operator\*(const box& b)

{

box box;

box.height = this->height \* b.height;

box.length = this->length \* b.length;

box.width = this->width \* b.width;

return box;

}

box operator/(const box& b)

{

box box;

box.height = this->height / b.height;

box.length = this->length / b.length;

box.width = this->width / b.width;

return box;

}

void showdata()

{

cout << "BOX HEIGHT IS : " << height << endl;

cout << "BOX LENGTH IS : " << length << endl;

cout << "BOX WIDTH IS : " << width << endl;

}

};

int main()

{

box b1(9.8, 7.6, 5.4);

box b2(2.3, 3.2, 1.2);

box b3;

b3 = b1 + b2;

b3.showdata();

cout << endl;

box b4;

b4 = b1 - b2;

b4.showdata();

cout << endl;

box b5;

b5 = b1 \* b2;

b5.showdata();

cout << endl;

box b6;

b6 = b1 / b2;

b6.showdata();

}

**OUTPUT :**

