//This is Lab2A

////comsc 200

//status: complete

// main.cpp

// lec2

// Created by Jeff on 8/22/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

#include <iostream>

#include <string>

using namespace std;

class Car{

private:

int year;

std::string make;

int speed;

public:

//constructor

Car(int y,std::string m){

year=y; make=m; speed=0;

}

Car(Car &c){

year = c.year;

make =c.make;

speed = c.speed;

}

// getter

int getYear(){return year;};

string getMake(){return make;}; // ??

int getSpeed(){return speed;};

//setter

void setYear(int y){year = y;}

int accelerate(){

if(speed <25){

return speed += 5;

}

else

return speed;

};

void brake(){

if(speed >=25){

while(speed !=0){

speed -=5;

std::cout<<"braking..... "<<"\n"<<"current speed: " << speed << ' ' << "\n";

}

}

else

{

std::cout << "accelerating " << "\n";

}

}

};

int main(){

Car porsche(2016,"porsche");

cout<<"my car Made by "<<porsche.getMake()<< ".\n";

cout<<"current speed : "

<<porsche.getSpeed()<<" mile per hour. \n";

for(int count=0;count<10;count++){

porsche.accelerate();

cout<<"current speed "<<porsche.getSpeed() << " mile per hour.\n";

porsche.brake();

}

}

//};



Lab 2 b

//comsc 200

//status: complete

#include <iostream>

#include <list>

#include <vector>

bool contains(std::list<int> mylist,int elem){

std::list<int>::iterator it;

for(it=mylist.begin();it!=mylist.end();++it)

if(\*it ==elem)return true;

return false;

}

int main ()

{

std::list<int> mylist;

std::list<int>::iterator it;

if(mylist.empty()) std::cout << "This newly created list contains nothing\n";

// set some initial values:

for (int i=1; i<=10; ++i) mylist.push\_back(i);

if(!mylist.empty()) std::cout << "This list contains something\n";

std::cout << "mylist contains:";

for (it=mylist.begin(); it!=mylist.end(); ++it)

std::cout << ' ' << \*it;

std::cout << '\n';

for(it=mylist.begin(); it!=mylist.end(); ++it)

{

if((\*it) %2 == 0){

it = mylist.erase(it);

it --;

}

}

std::cout << "mylist contains:";

for (it=mylist.begin(); it!=mylist.end(); ++it)

std::cout << ' ' << \*it;

std::cout << '\n';

for (int i=1; i<=3; ++i) mylist.push\_back(i);

std::cout << "mylist contains:";

for (it=mylist.begin(); it!=mylist.end(); ++it)

std::cout << ' ' << \*it;

std::cout << '\n';

for(int i=4;i<=6;i++){

std::cout<< "list contains: "<<i ;

if(contains(mylist,i))

std::cout << "yes" << '\n';

else

std::cout << "no" << '\n';

}

mylist.sort();

std::cout << "mylist contains:";

for (it=mylist.begin(); it!=mylist.end(); ++it)

std::cout << ' ' << \*it;

std::cout << '\n';

std::cout << "mylist has:" << mylist.size()<<"items. \n" ;

return 0;

}



lab 2 c

//

// main.cpp

// lab2c

//comsc 200

//status: complete

// Created by Jeff on 8/24/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

#include <iostream>

#include <string>

#include <vector>

using namespace std;

int main() {

string testString[6]{

"ABLE WAS I ERE I SAW ELBA",

"FOUR SCORE AND SEVEN YEARS AGO",

"NOW IS THE TIME FOR ALL GOOD MEN",

"DESSERTS I STRESSED",

"AKS NOT WHAT YOUR COUNTRY CAN DO FOR YOU",

"KAYAK",

};

for(int i=0;i<6;i++){

string rev = string(testString[i].rbegin(),testString[i].rend());

cout<< "\" "<<testString[i] << "\"" ;

if(testString[i].compare(rev)== 0)

cout <<"is a palinedrome. " << "\n" ;

else

cout << " is not a palinedrome" << endl;

}

}



lab 2 d

//

// main.cpp

// Lab2d

//comsc 200

//status: complete

// Created by Jeff on 8/22/16.

// Copyright © 2016 Jeff zhang. All rights reserved.

//

/\* The Circle class (All source codes in one file) (CircleAIO.cpp) \*/

#include <iostream> // using IO functions

#include <string> // using string

using namespace std;

class Circle {

private:

double radius; // Data member (Variable)

string color; // Data member (Variable)

public:

// Constructor with default values for data members

Circle(double r = 1.0, string c = "red") {

radius = r;

color = c;

}

double getRadius() { // Member function (Getter)

return radius;

}

string getColor() { // Member function (Getter)

return color;

}

double getArea() { // Member function

return radius\*radius\*3.1416;

}

}; // need to end the class declaration with a semi-colon

// Test driver function

int main() {

// Construct a Circle instance

Circle c1(1.2, "blue");

cout << "Radius=" << c1.getRadius() << " Area=" << c1.getArea()

<< " Color=" << c1.getColor() << endl;

// Construct another Circle instance

Circle c2(3.4); // default color

cout << "Radius=" << c2.getRadius() << " Area=" << c2.getArea()

<< " Color=" << c2.getColor() << endl;

// Construct a Circle instance using default no-arg constructor

Circle c3; // default radius and color

cout << "Radius=" << c3.getRadius() << " Area=" << c3.getArea()

<< " Color=" << c3.getColor() << endl;

return 0;

}

