COP4020 Programming Languages C++ Test 3

Edelis Molina

November 10, 2022

Contents

1 Problem 1

1.1 Rectangle.h

```
class Rectangle
{
private:
  float length;
  float width;

public:
  void setlength(float);
  void setwidth(float);
  float perimeter();
  float area();
  void show();
  int sameArea(Rectangle);
};
```

1.2 Rectangle.cpp

```
#include "Rectangle.h"
#include <iostream>
using namespace std;

void Rectangle::setlength(float len)
{
  length = len;
}

void Rectangle::setwidth(float wid)
{
  width = wid;
}

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

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

void Rectangle::show()
```

```
{
  cout << "Rectangle Length: " << length << endl
       << "Rectangle Width: " << width << endl;
}

int Rectangle::sameArea(Rectangle rec2)
{
  if (this->area() == rec2.area())
    return 1;
  return 0;
}
```

1.3 problem1.cpp

```
#include <iostream>
#include "Rectangle.h"
using namespace std;

int main()
{
  Rectangle rec1, rec2;
  rec1.setlength(5);
  rec1.setwidth(2.5);
  rec2.setlength(20);
  rec2.setwidth(2);

  cout << "========= First rectangle ========= : " << endl;
  rec1.show();
  cout << "Area: " << rec1.area() << endl
       << "Perimeter: " << rec1.perimeter() << endl;

  cout << "========= Second rectangle ========= : " << endl;
  rec2.show();
  cout << "Area: " << rec2.area() << endl
       << "Perimeter: " << rec2.perimeter() << endl;

  if (rec1.sameArea(rec2))
    cout << "Rectangles have the same area\n";
  else
    cout << "Rectangles do not have the same area\n";

  // set and show new dimensions
  rec1.setlength(10);
  rec1.setwidth(4);

  cout << "========= First rectangle ========= : " << endl;
  rec1.show();
  cout << "Area: " << rec1.area() << endl
       << "Perimeter: " << rec1.perimeter() << endl;

  cout << "========= Second rectangle ========= : " << endl;
  rec2.show();
  cout << "Area: " << rec2.area() << endl
       << "Perimeter: " << rec2.perimeter() << endl;

  if (rec1.sameArea(rec2))
    cout << "Rectangles have the same area\n";
  else
    cout << "Rectangles do not have the same area\n";

  return 0;
}
```

2 Problem 2

2.1 Complex.h

```
#include <iostream>
using namespace std;

class Complex
{
private:
  float real;
  float imaginary;

public:
  void set(float r, float img)
  {
    real = r;
    imaginary = img;
  }

  void disp()
  {
    cout << real << " + " << imaginary << "i" << endl;
  }

  Complex sum(Complex c)
  {
    Complex temp;
    temp.real = real + c.real;
    temp.imaginary = imaginary + c.imaginary;

    return temp;
  }
};
```

2.2 problem2.cpp

```
#include <iostream>
#include "Complex.h"
using namespace std;

int main()
{

  Complex c1, c2, c3;
  c1.set(4, 2);
  c2.set(5, 1);
  c3 = c1.sum(c2);

  cout << "Complex number 1:" << endl;
  c1.disp();
  cout << "Complex number 2:" << endl;
  c2.disp();
  cout << "Complex number 3 = Complex 1 + Complex 2:" << endl;
  c3.disp();

  return 0;
}
```

3 Problem 3

3.1 Distance.h

```
#include <iostream>
using namespace std;

class Distance
{
private:
  int feet;
  float inches;

public:
  void set(int f, float in)
  {
    feet = f;
    inches = in;
  }

  void disp()
  {
    cout << feet << " ft " << inches << " inches " << endl;
  }

  Distance add(Distance d)
  {
    Distance temp;
    temp.feet = feet + d.feet;
    temp.inches = inches + d.inches;

    // convert inches to feet if greater than 12
    while (temp.inches >= 12.0)
    {
      temp.inches = temp.inches - 12.0;
      ++temp.feet;
    }

    return temp;
  }
};
```

3.2 problem3.cpp

```
#include <iostream>
#include "Distance.h"

int main()
{
  Distance d1, d2, d3;
  d1.set(5, 4);
  d2.set(9, 20);
  d3 = d1.add(d2);

  cout << "Distance 1:" << endl;
  d1.disp();
  cout << "Distance 2:" << endl;
  d2.disp();
  cout << "Distance 3 = Distance 1 + Distance 2:" << endl;
  d3.disp();
  return 0;
}
```

4 Problem 4

4.1 Time.h

```
#include <iostream>
using namespace std;

class Time
{
private:
  int hours, minutes;

public:
  void settime(int h, int min)
  {
    hours = h;
    minutes = min;
  }

  void showtime()
  {
    cout << hours << " hours and " << minutes << " minutes" << endl;
  }

  Time sum(Time t)
  {
    Time temp;
    temp.hours = hours + t.hours;
    temp.minutes = minutes + t.minutes;

    // convert min to hours format if greater than 60
    while (temp.minutes > 60.0)
    {
      temp.minutes = temp.minutes - 60.0;
      temp.hours++;
    }
    // convert hours to 24 hours format if greater than 24
    while (temp.hours > 24.0)
    {
      temp.hours = temp.hours - 24.0;
    }
    return temp;
  }
};
```

4.2 problem4.cpp

```
#include <iostream>
#include "Time.h"

int main()
{
  Time t1, t2, t3;
  t1.settime(10, 50);
  t2.settime(15, 40);
  t3 = t1.sum(t2);

  cout << "Time 1:" << endl;
  t1.showtime();
  cout << "Time 2:" << endl;
  t2.showtime();
  cout << "Time 3 = Time 1 + Time 2:" << endl;
  t3.showtime();

  return 0;
}
```

5 Problem 5

5.1 CashRegister.h

```
class CashRegister
{
private:
  int cashOnHand;

public:
  // default amount of cash on the register
  CashRegister();

  // set cash on register to a different amount
  CashRegister(int cashIn);

  int getCurrentBalance();
  // Update amount in Register based on $ deposited by customer
  void acceptAmount(int amountIn);
};

CashRegister::CashRegister()
{
  cashOnHand = 500;
}

CashRegister::CashRegister(int cashIn)
{
  cashOnHand = cashIn;
}

void CashRegister::acceptAmount(int amountIn)
{
  cashOnHand += amountIn;
}

int CashRegister::getCurrentBalance()
{
  return cashOnHand;
}
```

5.2 DispenserType.h

```
class DispenserType
{
private:
  int numberOfItems;
  int cost;

public:
  // default constructor
  DispenserType();
  // overloaded constructor
  DispenserType(int setNumOfItems, int setCost);
  int getNoOfItems();
  int getCost();
  void makeSale();
};

DispenserType::DispenserType()
{
  numberOfItems = 50;
  cost = 50;
}
```

```
DispenserType::DispenserType(int setNumOfItems, int setCost)
{
  numberOfItems = setNumOfItems;
  cost = setCost;
}

int DispenserType::getNoOfItems()
{
  return numberOfItems;
}

int DispenserType::getCost()
{
  return cost;
}

void DispenserType::makeSale()
{
  numberOfItems--;
}
```

5.3 problem5.cpp

```
#include "CashRegister.h"
#include "DispenserType.h"
#include <iostream>
using namespace std;

void showSelection();
void sellProduct(DispenserType &, CashRegister &);

int main()
{
  // initialize vending machine
  DispenserType candy(100, 2);
  DispenserType chips(200, 3);
  DispenserType gum(300, 4);
  DispenserType cookies(50, 3);

  CashRegister regCounter;

  int ch;
  showSelection();
  cin >> ch;
  while (ch != 5)
  {
    switch (ch)
    {
    case 1:
      sellProduct(candy, regCounter);
      break;
    case 2:
      sellProduct(chips, regCounter);
      break;
    case 3:
      sellProduct(gum, regCounter);
      break;
    case 4:
      sellProduct(cookies, regCounter);
      break;
    default:
      cout << "Invalid selection." << endl;
    }
    showSelection();
    cin >> ch;
  }

  return 0;
}

void showSelection()
{
  cout << "**** Available Items in the Vending Machine ****" << endl;
  cout << "To select an item, enter: " << endl;
  cout << "1 for Candy" << endl;
  cout << "2 for Chips" << endl;
  cout << "3 for Gum" << endl;
  cout << "4 for Cookies" << endl;
  cout << "5 to exit" << endl;
  cout << "> ";
}

void sellProduct(DispenserType &product, CashRegister &pCounter)
{
  int amt;
  int extraAmt;

  // if there’s product type in the DispenserType
  if (product.getNoOfItems() > 0)
  {
    cout << "Item costs " << product.getCost() << " dollars" << endl;
    cout << "Please deposit " << product.getCost() << " to make the purchase or 0 to cancel
    cin >> amt;

    while (amt < product.getCost())
    {
      // sell is canceled
      if (amt == 0)
      {
        cout << "Sorry to see you go. " << endl;
        return;
      }

      cout << "Deposit an additional " << product.getCost() - amt << " dollars: ";
      cin >> extraAmt;
      amt += extraAmt;
    }

    if (amt >= product.getCost())
    {
      pCounter.acceptAmount(amt);
      product.makeSale();
      cout << "\nSale made successfully\n"
           << endl;
    }
  }
  else
  {
    cout << "Sorry, item is sold out" << endl;
  }
}
```