**15B17CI371 – Data Structures**

**Lab ODD 2024**

**Week 0-LAB B**

**Practice Lab**

1. **What does the following code do? Can it be used as a skeleton code which can then be modified and reused for solving different questions? Justify. #include using namespace std; class Sample { public: void printText1(); void printText2(); void printValue(int value); }; void Sample::printText1() { cout << "IncludeHelp.com\n"; } void Sample::printText2() { cout << "Let's learn together\n"; } void Sample::printValue(int value) { cout << "value is: " << value << "\n"; } int main() { Sample obj; obj.printText1(); obj.printText2(); obj.printValue(101);**

The given code defines a simple C++ class called Sample with three member functions (printText1, printText2, and printValue) and a main function that creates an instance of the Sample class and calls these member functions.

Yes, this code can be used as skeleton code which can then be modified and reused for solving different questions.

* It demonstrates basic member functions with different functionalities (printing text and values), which can be modified or extended to meet specific requirements.
* The use of a class and object creation is a fundamental concept in object-oriented programming, making it a good starting point for various problems.
* The member functions are modular and can be easily replaced or extended with new functionalities.

1. **Write a CPP program to design a calculator do the following: (i) Add two natural numbers. (ii) Add two complex numbers. (iii) Add two matrices. Your code must showcase the use of operator and/or function overloading.**

#include <iostream>

using namespace std;

int add(int a,int b)

{

return a+b;

}

class Complex

{

int r,c;

public :

int inp()

{

cout<<"Input the real part of the number : ";

cin>>r;

cout<<"Input the imaginary part of the number : ";

cin>>c;

return 0;

}

Complex operator +(Complex &obj)

{

Complex temp;

temp.r=r+obj.r;

temp.c=c+obj.c;

return temp;

}

void display()

{

if(c>=0)

cout<<"\nSum = "<<r<<" + "<<c<<"i";

else

cout<<"\nSum = "<<r<<" "<<c<<"i";

}

};

int main()

{

int num1,num2,r,c;

cin>>num1>>num2;

cout<<"Sum of two natural numbers: "<<add(num1,num2)<<endl;

Complex c1,c2,c3;

cout<<"First Complex Number :\n";

c1.inp();

cout<<"Second Complex Number :\n";

c2.inp();

c3=c1+c2;

c3.display();

cout<<"\nInput the size of the matrices : \n";

cin>>r>>c;

int mat1[r][c];

int mat2[r][c];

cout<<"Input values of Matrix 1 :\n";

for(int i=0;i<r;++i)

for(int j=0;j<c;++j)

cin>>mat1[i][j];

cout<<"Input values of Matrix 2 :\n";

for(int i=0;i<r;++i)

for(int j=0;j<c;++j)

cin>>mat2[i][j];

cout<<"Matrix 1 :\n";

for(int i=0;i<r;++i)

{

for(int j=0;j<c;++j)

cout<<mat1[i][j]<<" ";

cout<<endl;

}

cout<<"Matrix 2 :\n";

for(int i=0;i<r;++i)

{

for(int j=0;j<c;++j)

cout<<mat2[i][j]<<" ";

cout<<endl;

}

int result[r][c];

for(int i=0;i<r;++i)

for(int j=0;j<c;++j)

result[i][j]=mat1[i][j]+mat2[i][j];

cout<<"Sum of Matrix 1 and Matrix 2 :\n";

for(int i=0;i<r;++i)

{

for(int j=0;j<c;++j)

cout<<result[i][j]<<" ";

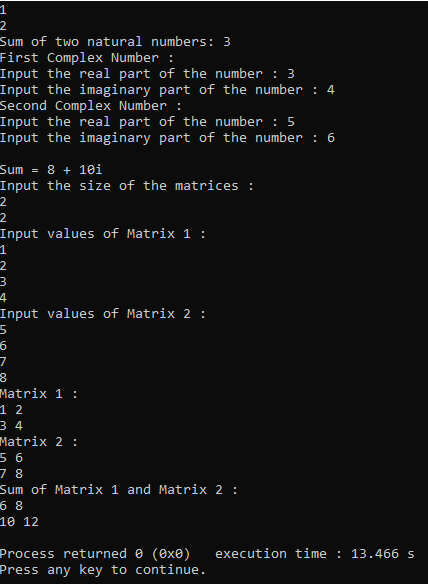
cout<<endl;

}

return 0;

}

**Output :**

****

1. **Write a CPP program to take input from user for 10 vendors who supply computer accessories. Design attributes and functions to satisfy the below mentioned requirements. Write function to (i) input accessory details of individual vendors. (ii) print the above details. (iii) Compare between the prices of the same component/accessory of different vendors. (iv) Find the vendor who has maximum quantity of “LAN Cable” currently available. (v) Find the vendor who has the lowest selling price of “Keyboard”.**

#include <iostream>

#include <string>

#include <vector>

#include <limits>

using namespace std;

class Accessory

{

public:

string name;

double price;

int quantity;

Accessory(string n="", double p=0.0, int q=0) : name(n), price(p), quantity(q) {}

};

class Vendor

{

public:

string name;

vector<Accessory> accessories;

Vendor(string n="") : name(n) {}

void inputAccessoryDetails()

{

int num;

cout<<"Enter number of accessories for vendor "<<name<<": ";

cin>>num;

accessories.clear();

for(int i=0;i<num;++i)

{

string accessoryName;

double accessoryPrice;

int accessoryQuantity;

cout<<"Enter accessory "<<i+1<<" details(name price quantity): ";

fflush(stdin);

getline(cin,accessoryName);

cin>> accessoryPrice>>accessoryQuantity;

accessories.push\_back(Accessory(accessoryName, accessoryPrice, accessoryQuantity));

}

}

void printDetails()

{

cout<<"Vendor: "<<name<<endl;

for(size\_t i=0;i<accessories.size();++i)

cout<<"Accessory: "<<accessories[i].name<<", Price: "<<accessories[i].price<<", Quantity: "<<accessories[i].quantity<<endl;

}

};

int main()

{

vector<Vendor> vendors(2);

for(int i=0;i<2;++i)

{

cout<<"Enter name for vendor "<<i+1<<": ";

string vendorName;

cin>>vendorName;

vendors[i]=Vendor(vendorName);

vendors[i].inputAccessoryDetails();

}

for(int i=0;i<2;++i)

{

vendors[i].printDetails();

cout<<endl;

}

string componentToCompare;

cout<<"Enter the name of the component to compare prices: ";

cin>>componentToCompare;

cout<<"Prices for "<<componentToCompare<<" from different vendors:"<<endl;

for(int i=0;i<2;++i)

for(size\_t j=0;j<vendors[i].accessories.size();++j)

if(vendors[i].accessories[j].name==componentToCompare)

cout<<"Vendor: "<<vendors[i].name<<", Price: "<<vendors[i].accessories[j].price<<endl;

string componentMaxQuantity="LAN Cable";

string maxVendor="";

int maxQuantity=0;

for(int i=0;i<2;++i)

for(size\_t j=0;j<vendors[i].accessories.size();++j)

if(vendors[i].accessories[j].name==componentMaxQuantity && vendors[i].accessories[j].quantity > maxQuantity)

{

maxQuantity=vendors[i].accessories[j].quantity;

maxVendor=vendors[i].name;

}

if(maxVendor != "")

cout<<"Vendor with maximum quantity of LAN Cable: "<<maxVendor<<", Quantity: "<<maxQuantity<<endl;

else

cout<<"No vendor has LAN Cable."<<endl;

string componentLowestPrice="Keyboard";

string minVendor="";

double minPrice=numeric\_limits<double>::max();

for(int i=0;i<2;++i)

for(size\_t j=0;j<vendors[i].accessories.size();++j)

if(vendors[i].accessories[j].name==componentLowestPrice && vendors[i].accessories[j].price<minPrice)

{

minPrice=vendors[i].accessories[j].price;

minVendor=vendors[i].name;

}

if(minVendor != "")

cout<<"Vendor with the lowest price of Keyboard: "<<minVendor<<", Price: "<<minPrice<<endl;

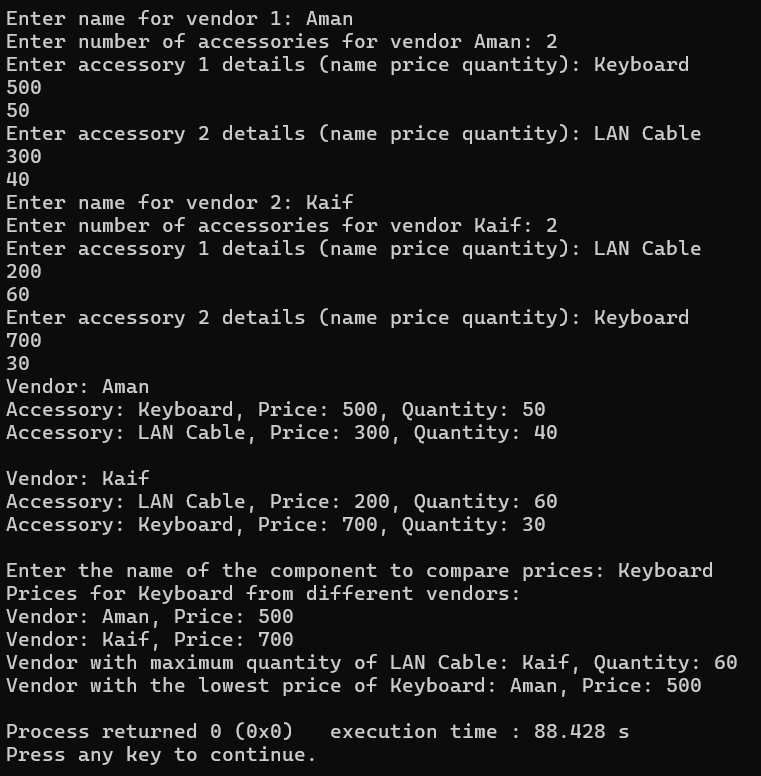
else

cout<<"No vendor has Keyboard."<<endl;

return 0;

}

**Output :**

****

1. Error : x is private within the test class
2. 1
3. Error : missing terminator “ character
4. Error : stray ‘\’ in program
5. Student's Roll No.: 0

Student's Name: None

Student's Percentage: 0

1. size of per: 1