**OBJECT ORIENTED PROGRAMMING LAB**

**LAB RECORD**

***Submitted by***

**SHASHANK MUDGAL**

**Submitted to: DR. Kunj Bihari Meena**



2022-2023

**Department of Computer Science & Engineering**

**JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, AB ROAD, RAGHOGARH, DT. GUNA-473226 MP, INDIA**

1. **The annual examination results of 10 students are tabulated as follows:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**roll No. Subject 1 Subject 2 Subject 3**

**. . . .**

**. . . .**

**. . . .**

**Write a program to read the data and determine the following:**

**a) Total marks obtained by each student.**

**b) The highest marks in each subject and the Roll No. of the students who secured it.**

**c) The student who obtained the highest total marks.**

**#include<iostream>**

**using namespace std;**

**class student{**

**int roll;**

**int s1,s2,s3;**

**int total;**

**public:**

**void setdata(){**

**cin>>roll>>s1>>s2>>s3;**

**}**

**void sum(){**

**total= s1+s2+s3;**

**}**

**friend void max(student[]);**

**};**

**void max(student a[]){**

**int max\_s1 = a[0].s1;**

**int temp = 0;**

**for(int i =1;i<10;i++){**

**if(max\_s1<a[i].s1){**

**max\_s1 = a[i].s1;**

**temp = i;**

**}**

**}**

**cout<<"topper in subject 1 is "<<a[temp].roll<<"marks secured "<<a[temp].s1<<endl;**

**int max\_s2 = a[0].s2;**

**temp = 0;**

**for(int i =1;i<10;i++){**

**if(max\_s2<a[i].s2){**

**max\_s2 = a[i].s2;**

**temp = i;**

**}**

**}**

**cout<<"topper in subject 2 is "<<a[temp].roll<<"marks secured "<<a[temp].s2<<endl;**

**int max\_s3 = a[0].s3;**

**temp = 0;**

**for(int i =1;i<10;i++){**

**if(max\_s3<a[i].s3){**

**max\_s3 = a[i].s3;**

**temp = i;**

**}**

**}**

**cout<<"topper in subject 3 is "<<a[temp].roll<<"marks secured "<<a[temp].s3<<endl;**

**int max\_total = a[0].total;**

**temp = 0;**

**for(int i =1;i<10;i++){**

**if(max\_total<a[i].total){**

**max\_total = a[i].total;**

**temp = i;**

**}**

**}**

**cout<<"topper in subject 3 is "<<a[temp].roll<<"marks secured "<<a[temp].total<<endl;**

**}**

**int main(){**

**student a[10];**

**cout<<"enter the details "<<endl;**

**for(int i = 0;i<10;i++){**

**a[i].setdata();**

**a[i].sum();**

**}**

**max(a[10]);**

**return 0;**

**}**

1. **Define a class to represent a bank account, including the following data members:**

* **Name of the depositor**
* **Account number**
* **Type of account**
* **Balance amount in the account**

**And member functions:**

* **To assign initial values**
* **To deposit an amount**
* **To withdraw an amount after checking the balance**
* **To display the name and balance**

**Write a main program to test the program.**

**Ans –//this programm is written by Shashank mudgal**

**#include <iostream>**

**using namespace std;**

**class bank**

**{**

**string name;**

**int account\_no;**

**string type;**

**int balance;**

**public:**

**void initial(void)**

**{**

**cout << "enter the account holder name " << endl;**

**cin >> name;**

**cout << "enter the account no " << endl;**

**cin >> account\_no;**

**cout << "enter the type of account " << endl;**

**cin >> type;**

**cout << "enter the balance in the account " << endl;**

**cin >> balance;**

**}**

**void deposit()**

**{**

**int amount;**

**cout << "enter the amount to deposit " << endl;**

**cin >> amount;**

**balance += amount;**

**}**

**void withdraw()**

**{**

**int draw;**

**cout << "enter the amount to withdraw " << endl;**

**cin >> draw;**

**if (draw > balance)**

**{**

**cout << "insufficient money you can't withdraw " << endl;**

**}**

**else**

**{**

**balance -= draw;**

**}**

**cout << "Re enter the amount to withdraw " << endl;**

**cin >> draw;**

**balance -= draw;**

**}**

**void display()**

**{**

**cout << "account holder name is " << name << endl;**

**cout << "updated balance is " << balance << endl;**

**}**

**};**

**int main()**

**{**

**bank a;**

**a.initial();**

**a.deposit();**

**a.withdraw();**

**a.display();**

**return 0;**

**}**

1. **Create two classes DM and DB which store the value of the distances. DM stores distances in meters and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. The object that stores the results may be a DM object or DB object, depending on the units in which the results are required. The display should be in the format of feet and inches or meters and centimeters depending on the object on display**.

**//This programm is written by shashank mudgal**

**#include <iostream>**

**using namespace std;**

**class db;**

**class dm**

**{**

**int feet;**

**int inches;**

**public:**

**void setdata(int a1, int a2)**

**{**

**feet = a1;**

**inches = a2;**

**}**

**friend void add(dm, db);**

**};**

**class db**

**{**

**int meter;**

**int centimeter;**

**public:**

**void setdata(int b1, int b2)**

**{**

**meter = b2;**

**centimeter = b1;**

**}**

**friend void add(dm, db);**

**};**

**void add(dm o1, db o2)**

**{**

**double d1,d2;**

**d1 = o2.meter+o1.feet/3.281;**

**d2 = o2.centimeter+o1.inches\*0.394;**

**cout<<"meter + feet = "<<d1<<" meter"<<endl;**

**cout<<"centimeter + inches = "<<d2<<" centimeter"<<endl;**

**}**

**int main()**

**{dm a;**

**db b;**

**a.setdata(1,2);**

**b.setdata(4,3);**

**add(a,b);**

**return 0;**

**}**