OOCP - Practical Assignment 1

Name: Vivekkumar Satishbhai jadav

Class: MCA 1st

Roll no: 9

1. **Write a program to create class Student with student’s rollno, name and marks of three subjects (OOCP, AI and MF) and display the details of student with total marks of all subjects along with the percentage in proper format.(Output should be in descending order of percentage.**

**// program**

#include<iostream>

#include<string>

using namespace std;

class Student{

private:

int rollno;

string name;

float OOCP, AI, MF, Marks, per;

public:

void inputData(){

cout<< "enter roll no :";

cin>> rollno;

cout<< "enter name :";

cin>> name;

//validation for marks not greater then 100

OOCP = inputMarks("OOCP");

AI = inputMarks("AI");

MF = inputMarks("MF");

calculateMarks();

}

// marks validation function

static float inputMarks(string subject){

float marks;

while (true) {

cout << "Enter marks for " << subject << " : ";

cin >> marks;

if (marks >= 0 && marks <= 100)

{

break;

}

else

{

cout << "Invalid! Please enter between 0 and 100.\n";

}

}

return marks;

}

// make differnt function for calculate

void calculateMarks() {

Marks = OOCP + AI + MF;

per = (Marks / 300) \* 100;

}

void displayData() {

cout << rollno << "\t" << name << "\t"

<< OOCP << "\t" << AI << "\t" << MF << "\t"

<< Marks << "\t" << per << "\n";

}

float getPercentage() {

return per;

}

};

int main() {

int n;

cout << "Enter number of students: ";

cin >> n;

Student students[100];

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

cout << "\n--- Student " << i + 1 << " ---\n";

students[i].inputData();

}

// Bubble Sort

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (students[j].getPercentage() < students[j + 1].getPercentage()) {

Student temp = students[j];

students[j] = students[j + 1];

students[j + 1] = temp;

}

}

}

cout << "\n\nRollNo\tName\tOOCP\tAI\tMF\tTotal\tPercentage\n";

cout << "-----------------------------------------------------------\n";

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

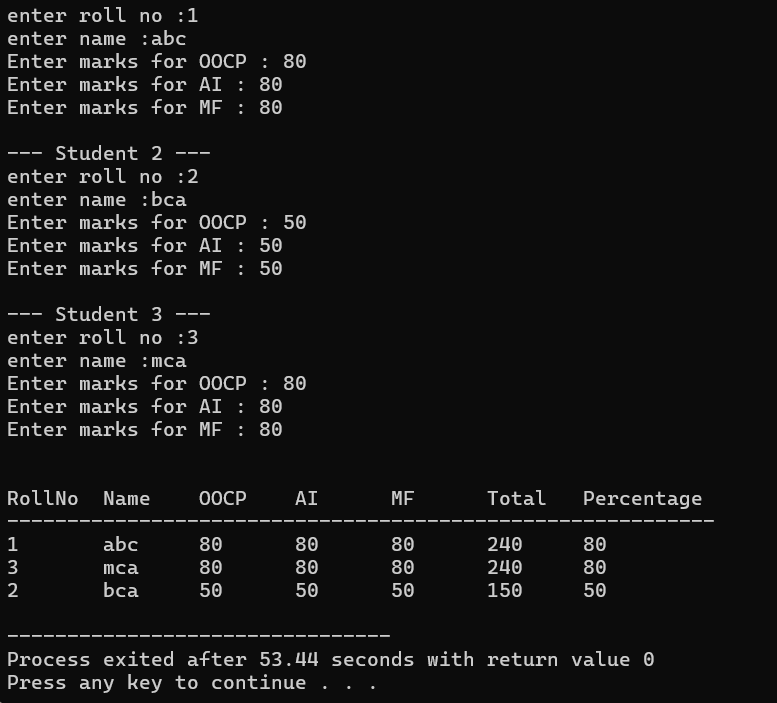
students[i].displayData();

}

return 0;

}

**// output**



1. **Write a program to create class Num (int n1, int n2, int n3, int n4). Display total and average of n1, n2, n3 and n4.**

**// program**

#include<iostream>

using namespace std;

class Num{

private:

int n1,n2,n3,n4,total;

float average;

public:

void input(){

cout<<"Enter n1:";

cin>>n1;

cout<<"Enter n2:";

cin>>n2;

cout<<"Enter n3:";

cin>>n3;

cout<<"Enter n4:";

cin>>n4;

}

void calculate(){

total = n1+n2+n3+n4;

average = total/4;

}

void display(){

cout<<"\n"<<" n1 is:"<<n1<<"\n"<<" n2 is:"<<n2<<"\n"<<" n3 is:"<<n3<<"\n"<<" n4 is:"<<n4<<endl;

cout<< "total:"<<total;

cout<<"average:"<<average;

}

};

int main(){

Num n;

n.input();

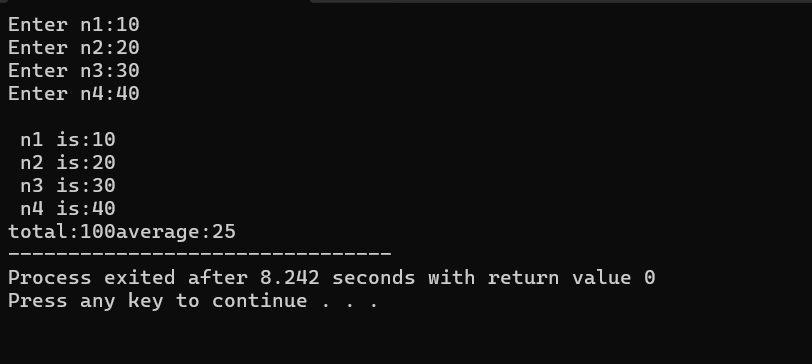
n.calculate();

n.display();

return 0;

}

**// output**



3. **Write a program to create class Time (int h, int m). Read a value as minutes from user to display new time after adding the value to minutes in Time.**

**// program**

#include<iostream>

using namespace std;

class Time{

private:

int h,m;

public:

//user input

void getData(){

cout << "Enter hours: ";

cin >> h;

cout << "Enter minutes: ";

cin >> m;

}

//add minute

void add(int extra) {

m += extra; // add minutes

h += m / 60; // change minutes to hours

m = m % 60; // make minute less then 60

}

void display() {

cout << "New Time = " << h << " hours " << m << " minutes" << endl;

}

};

int main() {

Time t;

int extra;

t.getData();

cout << "Enter extra minutes to add: ";

cin >> extra;

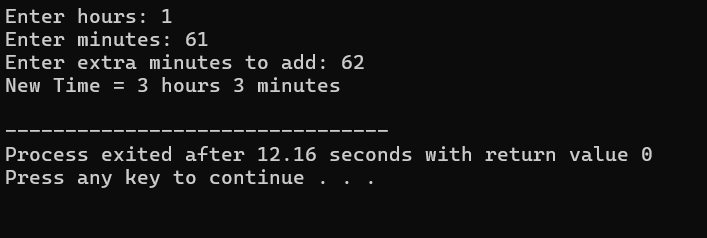
t.add(extra); // add minutes

t.display(); // show after add minute time

return 0;

}

**// output**



**4.** **Write a program to create class Date (int day, int month, int year). Read a value as day from user to display new date after adding the value to day in Date.**

**// program**

#include<iostream>

using namespace std;

class Date{

private:

int day ,month ,year;

int days(int m, int y) {

if (m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12)

return 31;

else if (m == 4 || m == 6 || m == 9 || m == 11)

return 30;

else if (m == 2) { // February

if ((y % 400 == 0) || (y % 4 == 0 && y % 100 != 0))

return 29; // leap year

else

return 28;

}

return 30;

}

public:

void input() {

cout << "Enter day: ";

cin >> day;

cout << "Enter month: ";

cin >> month;

cout << "Enter year: ";

cin >> year;

}

void addDays(int extra) {

day += extra;

// Adjust day, month, year

while (true) {

int maxDays = days(month, year);

if (day > maxDays) {

day -= maxDays;

month++;

if (month > 12) {

month = 1;

year++;

}

}

else {

break;

}

}

}

void display() {

cout << "New Date = " << day << "/" << month << "/" << year << endl;

}

};

int main() {

Date d;

int extra;

d.input();

cout << "Enter extra days to add: ";

cin >> extra;

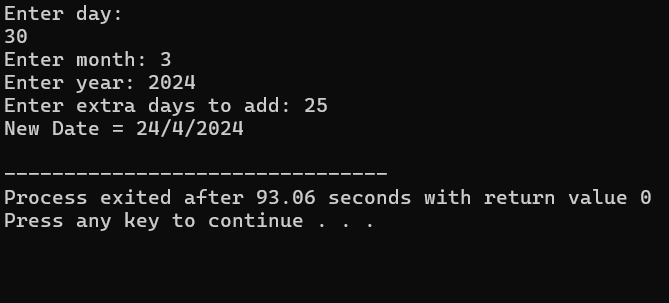
d.addDays(extra); // add days

d.display(); // display new date

return 0;

}

**// output**

****

**5. Write a program to create class employee with employee’s id, name and basic salary. Calculate gross salary for each employee(HRA 20%, DA 30%, OA 10%).**

**// program**

#include<iostream>

using namespace std;

class Employee{

private:

int id;

string name;

float basic ,gross,hra,da,oa;

public:

void input() {

cout << "Enter Employee ID: ";

cin >> id;

cout << "Enter Employee Name: ";

cin >> name;

cout << "Enter Basic Salary: ";

cin >> basic;

}

void calculate(){

hra = 0.20 \* basic;

da = 0.30 \* basic;

oa = 0.10 \* basic;

gross = basic + hra + da + oa;

}

void display() {

cout << "\nEmployee ID : " << id;

cout << "\nEmployee Name : " << name;

cout << "\nBasic Salary : " << basic;

cout << "\nHRA : " << hra;

cout << "\nDA : " << da;

cout << "\nOA : " << oa;

cout << "\nGross Salary : " << gross << endl;

}

};

int main() {

Employee e;

e.input(); // take employee details

e.calculate(); // calculate gross salary

e.display(); // show details

return 0;

}

**// output**

A screenshot of a computer

AI-generated content may be incorrect.

**6. Write a program to define a class called book. Write a program to read information about 10 books and display books details in ascending order of price in proper format.**

**// program**

#include <iostream>

#include<string>

using namespace std;

class Book {

private:

int id;

string title;

string author;

float price;

public:

void input() {

cout << "Enter Book ID: ";

cin >> id;

cin.ignore();

cout << "Enter Book Title: ";

getline(cin, title);

cout << "\nEnter Author Name: ";

getline(cin, author);

cout << "\nEnter Price: ";

cin >> price;

}

// take price

float getPrice() {

return price;

}

void display() {

cout << "Book Id:"<<id << "\t"<<"Book Title:" << title << "\t\t"<<"Book Author:" << author << "\t\t"<<"Book Price:" << price << endl;

}

};

int main() {

int SIZE;

cout << "How many books you need to add? :";

cin >> SIZE;

Book books[SIZE];

// input

cout << "Enter details of " << SIZE << " books:\n";

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

cout << "\nBook " << (i + 1) << ":\n";

books[i].input();

}

//sort

for (int i = 0; i < SIZE - 1; i++) {

for (int j = i + 1; j < SIZE; j++) {

if (books[i].getPrice() > books[j].getPrice()) {

swap(books[i], books[j]);

}

}

}

//dispaly

cout << "\nBooks in Ascending Order of Price:\n";

cout << "ID\tTitle\t\t\tAuthor\t\t\tPrice\n";

cout << "-----------------------------------------------------------\n";

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

books[i].display();

}

return 0;

}

**// output**

A screenshot of a computer program

AI-generated content may be incorrect.



**7.** **Demonstrate the use of static variables in a class by using it to count the number of times the value is being inputted in the program.**

**// program**

**#include<iostream>**

**using namespace std;**

**class Counter{**

**private:**

**int value;**

**static int count;**

**public:**

**void read(){**

**cout << "Enter a number: ";**

**cin >> value;**

**count++;// increase static value**

**}**

**void display() {**

**cout << "You entered: " << value << endl;**

**}**

**static void Count() {**

**// static function can access static members only**

**cout << "Total inputs taken: " << count << endl;**

**}**

**};**

**int Counter::count = 0;**

**int main() {**

**int n;**

**cout << "How many numbers you want to input? ";**

**cin >> n;**

**Counter obj[100]; // create array of objects**

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

**obj[i].read();**

**}**

**// show how many inputs taken**

**Counter::Count();**

**return 0;**

**}**

**// output**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**8.** **Create class STUDENT having rollno, name and age as data members, also take subject with three subjects and initialize their value with minimum passing marks. Using member function, modify marks of student with specific rollno which is given by user.**

**// program**

#include<iostream>

#include<string>

using namespace std;

class Student{

private:

int rollno;

string name;

int age;

int marks[3];

public:

//constructor

Student(int r=0, string n="" , int a=0){

rollno = r;

name = n;

age = a;

for (int i = 0; i < 3; i++) {

marks[i] = 35; // minimum pass mark is 35

}

}

void display(){

cout << "Roll No: " << rollno << ", Name: " << name << ", Age: " << age << endl;

cout << "Marks: ";

for (int i = 0; i < 3; i++) {

cout << marks[i] << " ";

}

cout << endl;

}

void change(int r) {

if (rollno == r) {

cout << "Enter new marks for 3 subjects of Roll No " << rollno << ": ";

for (int i = 0; i < 3; i++) {

cin >> marks[i];

}

cout << "Marks updated successfully!" << endl;

}

}

int getRollNo() {

return rollno;

}

};

int main() {

int n;

cout << "Enter number of students: ";

cin >> n;

Student s[100];

int roll; string name; int age;

// input student details

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

cout << "Enter Roll No:";

cin >> roll;

cout << "Enter Name:";

cin>>name;

cout << "Enter age:";

cin >> age;

s[i] = Student(roll, name, age);

}

// display students before change

cout << "\n--- Student Details Before change ---\n";

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

s[i].display();

}

// ask roll number to change marks

int searchRno;

cout << "\nEnter roll number of student to change marks: ";

cin >> searchRno;

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

s[i].change(searchRno);

}

// display students after change

cout << "\n--- Student Details After change ---\n";

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

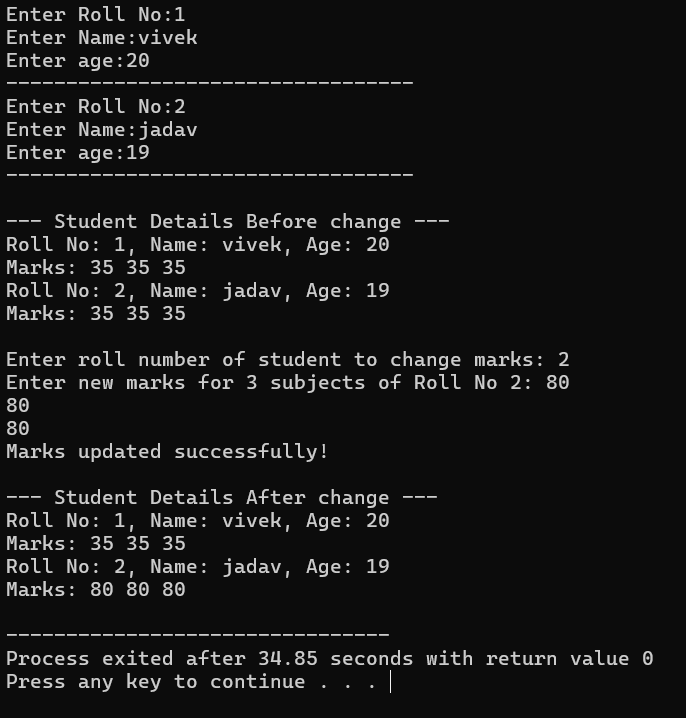
s[i].display();

}

return 0;

}

**// output**



**9. Define a class to represent a bank account. Include the following members :**

**DATA MEMBERS MEMBER FUNCTIONS**

**------------------------ ------------------------------**

**Name of depositor (1) To assign initial values**

**Account Number (2) To Deposit the amount**

**Type of Account (3) To withdraw an amount after checking the**

**Balance amount in account (4) To display name and balance**

**Write C++ program to handle 10 customers.**

**// program**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**class Bank {**

**private:**

**string name; // depositor name**

**int accno; // account number**

**string type; // account type**

**float bal; // balance**

**public:**

**// set initial values**

**void setdata(string n, int a, string t, float b) {**

**name = n;**

**accno = a;**

**type = t;**

**bal = b;**

**}**

**// deposit money**

**void deposit(float amt) {**

**bal = bal + amt;**

**cout << "Deposited: " << amt << endl;**

**cout << "Balance: " << bal << endl;**

**}**

**// withdraw money**

**void withdraw(float amt) {**

**if (amt > bal) {**

**cout << "Not enough balance!" << endl;**

**} else {**

**bal = bal - amt;**

**cout << "Withdrawn: " << amt << endl;**

**cout << "Balance: " << bal << endl;**

**}**

**}**

**// show account details**

**void display() {**

**cout << "\nName: " << name;**

**cout << "\nAcc No: " << accno;**

**cout << "\nType: " << type;**

**cout << "\nBalance: " << bal << endl;**

**}**

**// to search by account number**

**int getAccNo() {**

**return accno;**

**}**

**};**

**int main() {**

**Bank cust[10]; // 10 customers**

**// input for 10 customers**

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

**string n, t;**

**int a;**

**float b;**

**cout << "\nEnter details of customer " << i + 1 << ":\n";**

**cout << "Name: ";**

**cin.ignore(); // clear buffer**

**getline(cin, n);**

**cout << "Account No: ";**

**cin >> a;**

**cout << "Type (Saving/Current): ";**

**cin >> t;**

**cout << "Initial Balance: ";**

**cin >> b;**

**cust[i].setdata(n, a, t, b);**

**}**

**int ch, a;**

**do {**

**cout << "\n--- MENU ---\n";**

**cout << "1. Deposit\n";**

**cout << "2. Withdraw\n";**

**cout << "3. Display Account\n";**

**cout << "4. Exit\n";**

**cout << "Enter choice: ";**

**cin >> ch;**

**if (ch == 4) break;**

**cout << "Enter Account No: ";**

**cin >> a;**

**// find account**

**int pos = -1;**

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

**if (cust[i].getAccNo() == a) {**

**pos = i;**

**break;**

**}**

**}**

**if (pos == -1) {**

**cout << "Account not found!" << endl;**

**continue;**

**}**

**switch (ch) {**

**case 1: {**

**float amt;**

**cout << "Enter amount to deposit: ";**

**cin >> amt;**

**cust[pos].deposit(amt);**

**break;**

**}**

**case 2: {**

**float amt;**

**cout << "Enter amount to withdraw: ";**

**cin >> amt;**

**cust[pos].withdraw(amt);**

**break;**

**}**

**case 3:**

**cust[pos].display();**

**break;**

**default:**

**cout << "Invalid choice!" << endl;**

**}**

**} while (ch != 4);**

**cout << "\nThanks for using banking system!\n";**

**return 0;**

**}**

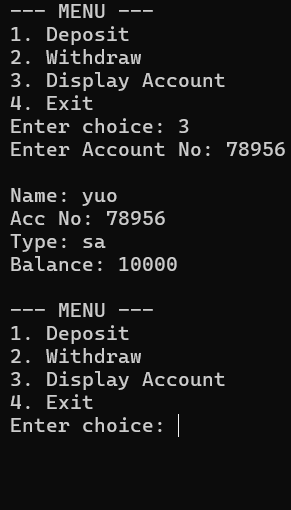
**// output**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

****

**10. Write a program to create class ‘Search’ having data members (int a[ ], x) and**

**define member functions as void input(), void output(), void search(int position),**

**void add(int value) to display result.**

**// program**

**#include <iostream>**

**using namespace std;**

**class Search {**

**private:**

**int a[50]; // array**

**int n; // size of array**

**public:**

**// take input**

**void input() {**

**cout << "Enter number of elements: ";**

**cin >> n;**

**cout << "Enter " << n << " elements:\n";**

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

**cin >> a[i];**

**}**

**}**

**// display array**

**void output() {**

**cout << "Array elements are: ";**

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

**cout << a[i] << " ";**

**}**

**cout << endl;**

**}**

**// search by position**

**void search(int pos) {**

**if (pos < 0 || pos >= n) {**

**cout << "Invalid position!" << endl;**

**} else {**

**cout << "Element at position " << pos << " = " << a[pos] << endl;**

**}**

**}**

**// add value at end**

**void add(int value) {**

**if (n < 50) {**

**a[n] = value;**

**n++;**

**cout << "Value " << value << " added successfully." << endl;**

**} else {**

**cout << "Array is full! Cannot add more elements." << endl;**

**}**

**}**

**};**

**int main() {**

**Search s;**

**s.input();**

**s.output();**

**int pos;**

**cout << "Enter position to search: ";**

**cin >> pos;**

**s.search(pos);**

**int val;**

**cout << "Enter value to add: ";**

**cin >> val;**

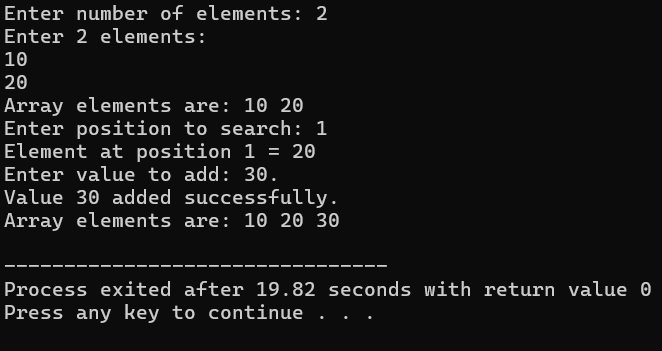
**s.add(val);**

**s.output(); // show updated array**

**return 0;**

**}**

**// output**

****