Roll no: 24

MCA: 1

Semester: 1

Name: Darshan Prajapati

// 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
{
   int rollno;
   string name;
   int marks_oocp;
   int marks_ai;
   int marks_mf;
   int total;

public:
   float percentage;
   // Take input of student's Details:
   void input ()
   {
      cout << "Enter roll no :";
      cin >> rollno;
```

```
    cout << "Enter name :";
    cin >> name;
    cout << "Enter marks of OOCP :";
    cin >> marks_oocp;

    cout << "Enter marks of AI :";
    cin >> marks_ai;
    cout << "Enter marks of MF :";
    cin >> marks_mf;
    cal();
}

// Calculate the total&percentage
void cal()
{
    total = marks_ai + marks_oocp + marks_mf;
    percentage = total / 3.0;
}

// display details of students in descending order by percentage
void display()
{
    cout << "\n";
    cout << "Your name is " << name << endl;
    cout << "Your rollno is " << rollno << endl;
    cout << "Your marks\n";
    cout << "OOCP:" << marks_oocp << endl;
    cout << "AI:" << marks_ai << endl;
    cout << "MF:" << marks_mf << endl;
    cout << "Total marks:" << total << endl;
    cout << "Percentage:" << percentage << "%" << endl;
```

```
    }
};

// function to sort students based on their percentage
void sortStudents(Student s[], int n)
{

    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (s[j].percentage < s[j + 1].percentage)
            {
                Student temp = s[j];
                s[j] = s[j + 1];
                s[j + 1] = temp;
            }
        }
    }
}

int main()
{
    int n;
    cout << "Enter total students:";
    cin >> n;
    Student *s = new Student[n];

    for (int i = 0; i < n; i++)
    {
        cout << "\nEnter details for student: " << i + 1 << "\n";
```

```
        s[i].input();
    }
    sortStudents(s, n);

    cout << "\nPrinting Students in descending order of percentage " << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "\nDetails of student: " << i + 1;
        s[i].display();
    }
    return 0;
}
```

// output:

// Enter details for student: 1
// Enter roll no :101
// Enter name :Darshan
// Enter marks of OOCP :45
// Enter marks of AI :65
// Enter marks of MF :89

// Enter details for student: 2
// Enter roll no :102
// Enter name :Meet
// Enter marks of OOCP :78
// Enter marks of AI :89
// Enter marks of MF :98

// Enter details for student: 3
// Enter roll no :103
// Enter name :Malav
// Enter marks of OOCP :89
// Enter marks of AI :74
// Enter marks of MF :56

// Printing Students in descending order of percentage

// Details of student: 1
// Your name is Meet
// Your rollno is 102
// Your marks
// OOCP:78
// AI:89
// MF:98
// Total marks:265
// Percentage:88.3333%

// Details of student: 2
// Your name is Malav
// Your rollno is 103
// Your marks
// OOCP:89
// AI:74
// MF:56
// Total marks:219
// Percentage:73%

// Details of student: 3
// Your name is Darshan
// Your rollno is 101
// Your marks
// OOCP:45
// AI:65
// MF:89
// Total marks:199
// Percentage:66.3333%

// 2. 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.

```
#include <iostream>
using namespace std;
class Num
{
   int n1;
   int n2;
   int n3;
```

```
    int n4;
    int total;
    int average;

public:
    // input values of numbers:
    void input()
    {
        cout << "Enter values of 4 numbers:" << endl;
        cin >> n1 >> n2 >> n3 >> n4;
    }
    void calculate()
    {
        total = n1 + n2 + n3 + n4;
        average = (total) / 4.0;
    }
    // display total and percentage:
    void display()
    {
        cout << "Total:" << total << endl;
        cout << "Average:" << average;
    }
};

int main()
{
    Num n;
    n.input();
    n.calculate();
    n.display();
    return 0;
}
```

// output:

// Enter values of 4 numbers:
// 42
// 12
// 52
// 68
// Total:174
// Average:43

// 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.

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

class Time
{
   int h;
   int m;

public:
   void input()
   {
      cout << "Enter hours:";
      cin >> h;
      cout << "Enter minutes:";
      cin >> m;
   }
   void addMinutes(int extra)
   {
      m += extra;  // total minutes = minutes+ extra
      h += m / 60; // total minutes/60 = add to hours
      m = m % 60;  // remining minutes
   }
   void display()
   {
      cout << "Total hours:" << h << "\n";
      cout << "Total minutes:" << m << "\n";
   }
};

int main()
{
   Time t;
   int extra;
   t.input();
   cout << "Enter extra minutes:";
   cin >> extra;
   t.addMinutes(extra);
```

```
    t.display();
    return 0;
}
```

// output:

// Enter hours:5
// Enter minutes:25
// Enter extra minutes:95
// Total hours:7
// Total minutes:0

// 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.

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

class Date
{
    int day;
    int month;
    int year;

public:
    void input()
    {
        cout << "Enter day:";
        cin >> day;

        cout << "Enter month:";
        cin >> month;

        cout << "Enter year:";
        cin >> year;
    }
    void addDays(int extra)
    {
        day += extra;
        // while (day > 30)
```

```
        // {
        //     day -= 30;
        //     month++;
        //     if (month > 12)
        //     {
        //         month = 1;
        //         year++;
        //     }
        // }
        // or //
        month += day / 30;
        day = day % 30;

        year += month / 12;
        month = month % 12;
        if (day == 0)
        {
            day = 30;
            month--;
            if (month == 0)
            {
                month = 12;
                year--;
            }
        }
    }
    void display()
    {
        cout << "\n";
        cout << "New Date: " << day << "/" << month << "/" << year << endl;
    }
};

int main()
{
    Date d;
    int extra;
    d.input();
    cout << "Enter extra days:";
    cin >> extra;
    d.addDays(extra);
```

```
    d.display();
    return 0;
}
```

// output:

// Enter day:5
// Enter month:12
// Enter year:1
// Enter extra days:25

// New Date: 30/12/1

// 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%).

```
#include <iostream>
#include <string>
using namespace std;
class Emp
{
    int id;
    string name;
    float basic_salary;
    float HRA = 0.20;
    float DA = 0.30;
    float OA = 0.10;

public:
    void input();
    void display();
};
void Emp ::input()
{
    cout << "Enter emp id:";
    cin >> id;
    cout << "Enter emp name:";
    cin >> name;
    cout << "Enter basic_salary:";
    cin >> basic_salary;
}
```

```
void Emp::display()
{
   float hra = basic_salary * HRA;
   float da = basic_salary * DA;
   float oa = basic_salary * OA;
   float gross_salary = basic_salary + hra + da + oa;

   cout << "\nEmployee ID: " << id;
   cout << "\nEmployee Name: " << name;
   cout << "\nBasic Salary: " << basic_salary;
   cout << "\nHRA: " << hra;
   cout << "\nDA: " << da;
   cout << "\nOA: " << oa;
   cout << "\nGross Salary: " << gross_salary << endl;
}
int main()
{
   Emp e;
   e.input();
   e.display();
   return 0;
}
```

// output:

// Enter emp id:101

// Enter emp name:Darshan

// Enter basic_salary:20000

// Employee ID: 101

// Employee Name: Darshan

// Basic Salary: 20000

// HRA: 4000

// DA: 6000

// OA: 2000

// Gross Salary: 32000

// 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.

```
#include <iostream>
```

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

class Book
{
   int id;
   string title;

public:
   float price;
   void input()
   {
      cout << "Enter Book_Id:";
      cin >> id;
      cout << "Enter Title:";
      cin >> title;
      cout << "Enter Price:";
      cin >> price;
   }
   void display()
   {
      cout << "\tID: " << id << "\tTitle: " << title << "\tPrice: " << price;
   }
};
// sort books in asceding order as per price
void sortBooks(Book b[], int n)
{
   for (int i = 0; i < n - 1; i++)
   {
      for (int j = 0; j < n - i - 1; j++)
      {
         if (b[j].price > b[j + 1].price)
         {
            Book temp = b[j];
            b[j] = b[j + 1];
            b[j + 1] = temp;
         }
      }
   }
}
int main()
```

```
{
   int n = 10;
   Book b[n];
   // input
   cout << "Enter Book Details";
   for (int i = 0; i < n; i++)
   {
      cout << "\nBook " << i + 1 << "\n";
      b[i].input();
   }

   // sortbooks
   sortBooks(b, n);

   // display
   cout << "Book Details in asceding order" << endl;
   for (int i = 0; i < n; i++)
   {
      cout << "\n Book: " << i + 1 << "|";
      b[i].display();
      ;
   }
   return 0;
}
```

// output:

// Enter Book Details
// Book 1
// Enter Book_Id:101
// Enter Title:book1
// Enter Price:1123

// Book 2
// Enter Book_Id:102
// Enter Title:book2
// Enter Price:1299

// Book 3
// Enter Book_Id:103
// Enter Title:book3
// Enter Price:2399

// Book 4
// Enter Book_Id:104
// Enter Title:book4
// Enter Price:9899

// Book 5
// Enter Book_Id:105
// Enter Title:book5
// Enter Price:9999

// Book 6
// Enter Book_Id:106
// Enter Title:book6
// Enter Price:8799

// Book 7
// Enter Book_Id:107
// Enter Title:book7
// Enter Price:799

// Book 8
// Enter Book_Id:108
// Enter Title:book8
// Enter Price:399

// Book 9
// Enter Book_Id:109
// Enter Title:book9
// Enter Price:499

// Book 10
// Enter Book_Id:110
// Enter Title:book10
// Enter Price:699
// Book Details in asceding order

// Book: 1| ID: 108 Title: book8 Price: 399
// Book: 2| ID: 109 Title: book9 Price: 499
// Book: 3| ID: 110 Title: book10 Price: 699
// Book: 4| ID: 107 Title: book7 Price: 799

// Book: 5| ID: 101 Title: book1 Price: 1123
// Book: 6| ID: 102 Title: book2 Price: 1299
// Book: 7| ID: 103 Title: book3 Price: 2399
// Book: 8| ID: 106 Title: book6 Price: 8799
// Book: 9| ID: 104 Title: book4 Price: 9899
// Book: 10| ID: 105 Title: book5 Price: 9999

// 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.

```
#include <iostream>
using namespace std;
class Counter
{
   int val;
   static int count;

public:
   void input()
   {
      count++;
      cout << "Enter value :";
      cin >> val;
   }
   void display()
   {
      cout << "value :" << val << endl;
   }
   static void countValue()
   {
      cout << "Function is called " << count << " Times";
   }
};
// intialized variable
int Counter::count = 0;
int main()
{
   int n;
   cout << "How many values you want to enter? ";
   cin >> n;
   Counter c[n];
```

```
    for (int i = 0; i < n; i++)
    {
        c[i].input();
        c[i].display();
    }

    Counter::countValue();

    return 0;
}
```

// output:

// How many values you want to enter? 5
// Enter value :12
// value :12
// Enter value :52
// value :52
// Enter value :14
// value :14
// Enter value :25
// value :25
// Enter value :12
// value :12
// Function is called 5 Times

// 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.

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

class Student
{

public:
    int rollno;
    string name;
    int age;
```

```
    int sub[3];
    // initialize minimum passing marks
    Student()
    {
        for (int i = 0; i < 3; i++)
        {
            sub[i] = 35;
        }
    }

    void input()
    {
        cout << "Enter rollno:";
        cin >> rollno;
        cout << "Enter name:";
        cin >> name;
        cout << "Enter age:";
        cin >> age;
    }
    void display()
    {
        cout << "\t Rollno: " << rollno << "\t Name: " << name << "\t Age:" << age << "\t Marks:";
        for (int i = 0; i < 3; i++)
        {
            cout << sub[i] << " ";
        }
        cout << " " << endl;
    }
};
// modifymarks using [roll no , student array of objects and numbers of students]:
void modifyMarks(int rollno, Student s[], int n)
{
    for (int i = 0; i < n; i++)
    {
        if (s[i].rollno == rollno)
        {
            cout << "Enter new marks for 3 subjects:\n ";
            {
                for (int j = 0; j < 3; j++)
                {
                    cin >> s[i].sub[j];
```

```
            }
            cout << "Marks updated successfully!\n";
            return;
          }
        }
    }
    cout << "Roll number not found!\n";
}

int main()
{
    int n;
    cout << "How many students you want to store:";
    cin >> n;
    Student s[n];
    // input for students
    for (int i = 0; i < n; i++)
    {
        s[i].input();
    }
    // modify marks
    int no;
    cout << "Enter roll no to be updated: ";
    cin >> no;
    modifyMarks(no, s, n);

    // display
    cout << "display details:\n";
    for (int i = 0; i < n; i++)
    {
        s[i].display();
    }

    return 0;
}
```

// output:

// How many students you want to store:2
// Enter rollno:101
// Enter name:Darshan

// Enter age:22
// Enter rollno:102
// Enter name:Kp
// Enter age:23
// Enter roll no to be updated: 102
// Enter new marks for 3 subjects:
// 78
// 87
// 89
// Marks updated successfully!
// display details:
// Rollno: 101 Name: Darshan Age:22 Marks:35 35 35
// Rollno: 102 Name: Kp Age:23 Marks:78 87 89

// 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.

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

class BankAccount
{
   string name;
   int accNo;
   string type;
   double balance;

public:
   // (1) To assign initial values
   BankAccount(string n = "", int a = 0, string t = "", double b = 0.0)
   {
```

```
   name = n;
   accNo = a;
   type = t;
   balance = b;
}
// input from user
void input()
{
   cout << "Enter your name:";
   cin >> name;
   cout << "Enter your account number:";
   cin >> accNo;
   cout << "What is acc type? ";
   cin >> type;
   cout << "Enter your account balance:";
   cin >> balance;
}
// (2) To Deposit the amount
void deposit(double amt)
{
   balance += amt;
   cout << "\tDeposited:" << amt << "\tTotal balance:" << balance << endl;
   display();
}
// (3) To withdraw an amount after checking the
void withdraw(double amt)
{
   if (amt > balance)
   {
      cout << "Insuffient balance...";
      return;
   }
   else
   {
      balance -= amt;
      cout << "\tWithdrawal:" << amt << "\tTotal balance:" << balance << endl;
      display();
   }
}
// (4) To display name and balance
void display()
```

```
    {
        cout << "\t Name:" << name << "\tAccount no:" << accNo << "\t Type of ac:" << type << "\tBalance:"
<< balance << endl;
    }
};
int main()
{
    int n = 10;
    BankAccount *cust = new BankAccount[n];
    cout << "Enter Details of Customers:\n";
    for (int i = 0; i < n; i++)
    {
        cout << "Customer: " << i + 1 << "\n";
        cust[i].input();
    }
    cout << "\n Details of Customers:\n";
    for (int i = 0; i < n; i++)
    {
        cout << "Customer: " << i + 1;
        cust[i].display();
    }
    cout << "\n";

    // deposit
    cust[0]
        .deposit(500);

    cout << "\n";
    // withdraw
    cust[1].withdraw(1000);

    delete[] cust;

    return 0;
}
```

// output:

// Enter Details of Customers:
// Customer: 1
// Enter your name:dp
// Enter your account number:101

// What is acc type? saving
// Enter your account balance:1200
// Customer: 2
// Enter your name:kp
// Enter your account number:1200
// What is acc type? current
// Enter your account balance:1250
// Customer: 3
// Enter your name:Garv
// Enter your account number:781
// What is acc type? current
// Enter your account balance:8999
// Customer: 4
// Enter your name:happy
// Enter your account number:1250
// What is acc type? current
// Enter your account balance:78000
// Customer: 5
// Enter your name:Taksh
// Enter your account number:878
// What is acc type? current
// Enter your account balance:125421
// Customer: 6
// Enter your name:Rahul
// Enter your account number:450
// What is acc type? current
// Enter your account balance:1212100
// Customer: 7
// Enter your name:umang
// Enter your account number:878
// What is acc type? Saving
// Enter your account balance:78200
// Customer: 8
// Enter your name:Yash
// Enter your account number:787
// What is acc type? Current
// Enter your account balance:1250
// Customer: 9
// Enter your name:Harsh
// Enter your account number:787
// What is acc type? Saving

// Enter your account balance:1200
// Customer: 10
// Enter your name:Dp
// Enter your account number:411
// What is acc type? Saving
// Enter your account balance:120000

// Details of Customers:
// Customer: 1 Name:dp Account no:101 Type of ac:saving Balance:1200
// Customer: 2 Name:kp Account no:1200 Type of ac:current Balance:1250
// Customer: 3 Name:Garv Account no:781 Type of ac:current Balance:8999
// Customer: 4 Name:happy Account no:1250 Type of ac:current Balance:78000
// Customer: 5 Name:Taksh Account no:878 Type of ac:current Balance:125421
// Customer: 6 Name:Rahul Account no:450 Type of ac:current Balance:1.2121e+006
// Customer: 7 Name:umang Account no:878 Type of ac:Saving Balance:78200
// Customer: 8 Name:Yash Account no:787 Type of ac:Current Balance:1250
// Customer: 9 Name:Harsh Account no:787 Type of ac:Saving Balance:1200
// Customer: 10 Name:Dp Account no:411 Type of ac:Saving Balance:120000

// Deposited:500 Total balance:1700
// Name:dp Account no:101 Type of ac:saving Balance:1700

// Withdrawal:1000 Total balance:250
// Name:kp Account no:1200 Type of ac:current Balance:250

// 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

```
#include <iostream>
using namespace std;
class Search
{
   int a[100];
   int n;
   int x;

public:
   void input();
   void output();
```

```
    void search(int position);
    void add(int value);
};

// input
void Search::input()
{
    cout << "Enter size of array:";
    cin >> n;

    cout << "\nEnter values of array element:\n";
    for (int i = 0; i < n; i++)
    {
        cin >> a[i];
    }

    cout << "Enter element to be search:";
    cin >> x;
}

// output
void Search::output()
{
    cout << "\nvalues of array element:\n";
    for (int i = 0; i < n; i++)
    {
        cout << a[i] << " ";
    }
}

// add
void Search::add(int value)
{
    a[n] = value;
    n++;
}
// search
void Search::search(int position)
{
    if (a[position] == x)
    {
```

```
      cout << "Element " << x << " is found at position " << position;
   }
   else
   {
      cout << "Element " << x << " not found! at position " << position;
   }
}
int main()
{
   Search s;
   s.input();
   s.output();
   s.add(50);
   cout << "\nAfter adding new element \n";
   s.output();
   int pos;
   cout << "\nEnter pos : ";
   cin >> pos;
   s.search(pos);
   return 0;
}
```

// output:

// Enter size of array:5

// Enter values of array element:
// 25
// 23
// 63
// 45
// 8
// Enter element to be search:8

// values of array element:
// 25 23 63 45 8
// After adding new element

// values of array element:
// 25 23 63 45 8 50
// Enter pos : 4
// Element 8 is found at position 4