Department Of Computer Science

Gujarat University

Certificate

***R**oll No:* *40* *Seat No:* _____

This is to certify that Mr. / Ms. ***Akshit Trivedi Ajaybhai*** *student of MCA Semester - I, has duly completed his/her term work for the semester ending in February 2022, in the subject of* ***Object Oriented Concepts And Programming*** *towards partial fulfillment of his / her Degree of Masters in Computer Science & Application.*

25/02/2022
Date of Submission

Internal Faculty

Head of Department

Department Of Computer Science
Gujarat University

MCA – 1

Subject: - **Object Oriented Concepts and Programming**

Name: - **Akshit Trivedi Ajaybhai**

Roll No.: - **40** **Exam Seat No.: -**

Sr. No.	Contents	Pg. No	Date	Signature
1	Assignment-1	3	21/02/2022	
2	Assignment-2	34	21/02/2022	
3	Assignment-3	53	21/02/2022	

Rollwala Computer Center

OBJECT ORIENTED CONCEPTS AND PROGRAMMING

Assignment-1

Feb 02, 2022

Name: **Akshit Trivedi**

Roll No: **40**

Course: **Master of Computer Application**

Sem: **1**

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

1. **Write a program that reads a character and prints out whether it is a vowel or a consonant or a number or any other character.**

INPUT:

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

int main()
{
    char c;
    cout<<"Enter any  Character: ";
    cin>>c;
    c=(char)tolower(c);
    cout<<"Character Entered by you is: "<<c<<endl;
    if(isdigit(c))
    {
        cout<<"\n"<<c<<" is a Digit.";
    }
    else if(isalpha(c))
    {
        if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u')
        {
            cout<<"\n"<<c<<" is a Vowel.";
        }
        else
        {
            cout<<"\n"<<c<<" is a Constant.";
        }
    }
    else
    {
        cout<<"\n"<<c<<" is a Special Character.";
    }
    return 0;
}
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**
Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 1\1_Constant_Vowel.exe
Enter any Character: a
Character Entered by you is: a

a is a Vowel.
```

```
C:\MCA\MCA-1\OOCP\Practical Assignment 1\1_Constant_Vowel.exe
Enter any Character: *
Character Entered by you is: *

* is a Special Character.
```

```
C:\MCA\MCA-1\OOCP\Practical Assignment 1\1_Constant_Vowel.exe
Enter any Character: 8
Character Entered by you is: 8

8 is a Digit.
```

2. WAP to add an 8% tax to a given amount and round the net amount to its positive nearest amount.

INPUT:

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


int main()
{
    float amt, tax;
    int tot_amt;
    cout<<"Enter the Amount: ";
    cin>>amt;
```

```
    tax = amt * 0.08;
    tot_amt = int(amt + tax);
    cout<<"The Tax on amount: "<< amt << " is " << tax << endl;
    cout<<"The Total amount is: "<< tot_amt;
    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 1\2_Tax.exe
Enter the Amount: 25000
The Tax on amount: 25000 is 2000
The Total amount is: 27000
```

4. WAP that takes a series of numbers and counts the frequency of positive values and negative values.

INPUT:

```
#include<iostream>
using namespace std;
int main()
{
    int pos=0,neg=0,zero=0,arr[100],i,arr_size;
    cout<<"Enter the How Many Elements you want to Enter (Max 100): ";
    cin>>arr_size;
    cout<<"\nEnter "<<arr_size<<" Elements: ";
    for(i=0;i<arr_size;i++)
    {
      cout<<"\nEnter Element "<<i+1<<" : ";
        cin>>arr[i];
```

```
        if(arr[i]<0)
        {
            neg++;
        }
        else if(arr[i]==0)
        {
            zero++;
        }
        else
        {
            pos++;
        }
    }
    cout<<"\nFrequency of Positive Numbers: "<<pos;
    cout<<"\nFrequency of Negative Numbers: "<<neg;
    cout<<"\nFrequency of Zero: "<<zero;
    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 1\4_Count_Positive_Negative.exe
Enter the How Many Elements you want to Enter (Max 100): 5

Enter 5 Elements:
Enter Element 1 : -5

Enter Element 2 : 40

Enter Element 3 : -16

Enter Element 4 : 0

Enter Element 5 : 21

Frequency of Positive Numbers: 2
Frequency of Negative Numbers: 2
Frequency of Zero: 1
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

5. WAP to convert the distance in meter to centimeter and feet to inches and vice versa using class DISTANCE. (1 meter = 100 centimeter and 1 feet = 12 inches).

INPUT:

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

class Distance
{
    public:
    float met_to_cen(float meter)
    {
        return  meter * 100;
    }

    float cen_to_met(float centi)
    {
        return centi / 100;
    }

    float feet_to_inch(float feet)
    {
        return feet * 12;
    }

    float inch_to_feet(float inch)
    {
```

```
        return inch / 12;
    }
};

int main()
{
    Distance d1;
    int op = 0;
    float item;
    while(op != 5)
    {
        cout<<"\n1: Convert from METRES To CENTIMETRES"<<endl;
        cout<<"2: Convert from CENTIMETRES To METRES"<<endl;
        cout<<"3: Convert from FEET To INCHES"<<endl;
        cout<<"4: Convert from INCHES To FEET"<<endl;
        cout<<"5: Exit"<<endl;
        cout<<"Enter your choice: ";
        cin>>op;

            cout<<"\n";
        switch(op)
        {
            case 1: cout<<"Enter the input in meter: ";
                    cin>>item;
                    cout<<item <<" Metres = " << d1.met_to_cen(item) << "
Centimetres"<< endl;
                    break;
            case 2: cout<<"Enter the input in centimeter: ";
```

```
                    cin>>item;
                    cout<< item <<" Centimetres = " << d1.cen_to_met(item) <<
" Metres"<< endl;
                    break;
            case 3: cout<<"Enter the input in feet: ";
                    cin>>item;
                    cout<< item <<" Feet = " << d1.feet_to_inch(item) << "
Inches"<< endl;
                    break;
            case 4: cout<<"Enter the input in inches: ";
                    cin>>item;
                    cout<< item <<" Inches = " << d1.inch_to_feet(item) << "
Feet"<< endl;
                    break;
            case 5: cout<<"Exiting the program" << endl;
                    break;
            default : cout<<"Enter a valid choice" << endl;
                    break;
        }
    }
}
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
Select C:\MCA\MCA-1\OOCP\Practical Assignment 1\5_Distance_Convertor.exe

1: Convert from METRES To CENTIMETRES
2: Convert from CENTIMETRES To METRES
3: Convert from FEET To INCHES
4: Convert from INCHES To FEET
5: Exit
Enter your choice: 1

Enter the input in meter: 153
153 Metres = 15300 Centimetres

1: Convert from METRES To CENTIMETRES
2: Convert from CENTIMETRES To METRES
3: Convert from FEET To INCHES
4: Convert from INCHES To FEET
5: Exit
Enter your choice: 2

Enter the input in centimeter: 15300
15300 Centimetres = 153 Metres

1: Convert from METRES To CENTIMETRES
2: Convert from CENTIMETRES To METRES
3: Convert from FEET To INCHES
4: Convert from INCHES To FEET
5: Exit
Enter your choice: 3

Enter the input in feet: 123
123 Feet = 1476 Inches

1: Convert from METRES To CENTIMETRES
2: Convert from CENTIMETRES To METRES
3: Convert from FEET To INCHES
4: Convert from INCHES To FEET
5: Exit
Enter your choice: 4

Enter the input in inches: 1476
1476 Inches = 123 Feet

1: Convert from METRES To CENTIMETRES
2: Convert from CENTIMETRES To METRES
3: Convert from FEET To INCHES
4: Convert from INCHES To FEET
5: Exit
Enter your choice: 5

Exiting the program
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

6. Create a class for Bank account with the following data members.

(1) Name of depositor

(2) Account number

(3) Type of account

(4) Balance member functions

a. To assign initial values

b. To deposit an amount in a particular account

c. To withdraw an amount after checking the balance

d. To display name and balance

WAP to manage at least 10 customers who can deal with deposit and withdraw amount and calculate the current balance.

INPUT:

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

class Account
{
    //data members
    static int totAccount;
    char name[20];
    int account_no;
    char account_type[20];
    float bal=0;

    public:
    Account()
    {
```

```
        account_no = 1000 + totAccount;
        totAccount++;
        bal=0;
    }

    //member function
    void deposit();
    void withdraw();
    void show_bal();
    void addAccount();

};

int Account::totAccount=0;

void Account::addAccount()
{
        int choice=0;
        cout<<"Enter your name: ";
        cin>>name;

        while(choice != 1 && choice != 2)
        {
            cout<<"Please choose your Account Types"<<endl;
            cout<<"1: Saving Account"<<endl;
            cout<<"2: Current Account"<<endl;
            cout<<"Enter your choice: ";
            cin>>choice;
```

```
            switch(choice)
            {
                case 1: strcpy(account_type, "Saving Account");
                    break;
                case 2: strcpy(account_type, "Current Account");
                    break;
                default: cout<<"Please enter a valid choice"<<endl;
                    break;
            }

        }
            cout<<"\n\nFinal details:"<<endl;
            cout<<"Account Number: "<<account_no<<endl;
            cout<<"Name: "<<name<<endl;
            cout<<"Account Type: "<<account_type<<endl;
            cout<<"Your initial balance is 0.Please deposit into your
account\n\n";

}

void Account::deposit()
{
    int cash=0;
    cout<<"Enter the amount: ";
    cin>>cash;
    bal = bal + cash;
    cout<<"The balance has been updated"<<endl;
```

```
}


void Account::withdraw()
{
    int cash=0;
    cout<<"Enter the amount: ";
    cin>>cash;
    if(bal==0 || cash>bal)
    {
        cout<<"You don't have sufficient balance to withdraw the given
amount"<<endl;
        return;
    }
    bal = bal - cash;
    cout<<"Your current balance is: "<<bal<<endl;
}


void Account::show_bal()
{
    cout<<"Account_no: "<< account_no << endl;
    cout<<"Name: "<< name << endl;
    cout<<"Total Balance: " << bal<<endl;
}


int check_account_no()
{
    int account_no;
    cout<<"Enter your account number: ";
```

```
    cin>>account_no;
    int j = account_no % 1000;
    if(j>10 || j<0 || account_no > 1011 || account_no < 1000 )
    {
        cout<<"Invalid Account Number"<<endl;
        return -1;
    }
    return j;
}

int main()
{
    int i=0;
    Account account[5];
    int accno, j;
    int op;
    while(op!=5)
    {
        cout<<"\n1: Add a new Account"<<endl;
        cout<<"2: Deposit"<<endl;
        cout<<"3: Withdraw"<<endl;
        cout<<"4: Show details"<<endl;
        cout<<"5: Exit"<<endl;
        cout<<"Enter your choice: ";
        cin>>op;
        switch(op)
        {
            case 1:
```

```
                if(i==5)
                {
                    cout<<"Users are exceeded"<<endl;
                    break;
                }
                account[i++].addAccount();
                break;

        case 2:
                j = check_account_no();
                if(j==-1)
                {
                    break;
                }
                account[j].deposit();
                break;

        case 3:
                j = check_account_no();
                if(j==-1)
                {
                    break;
                }
                account[j].withdraw();
                break;

        case 4:
                j = check_account_no();
```

```
                    if(j==-1)
                    {
                        break;
                    }
                    account[j].show_bal();
                    break;

            case 5: cout<<"Exiting!!!"<<endl;
                    break;

            default: cout<<"Please enter a valid choice";
                    break;
        }
    }
    return 0;
}
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

C:\MCA\MCA-1\OOCP\Practical Assignment 1\6_Bank_Account.exe

```
1: Add a new Account
2: Deposit
3: Withdraw
4: Show details
5: Exit
Enter your choice: 1
Enter your name: Akshit
Please choose your Account Types
1: Saving Account
2: Current Account
Enter your choice: 1


Final details:
Account Number: 1000
Name: Akshit
Account Type: Saving Account
Your initial balance is 0.Please deposit into your account


1: Add a new Account
2: Deposit
3: Withdraw
4: Show details
5: Exit
Enter your choice: 2
Enter your account number: 1000
Enter the amount: 5000
The balance has been updated

1: Add a new Account
2: Deposit
3: Withdraw
4: Show details
5: Exit
Enter your choice: 4
Enter your account number: 1000
Account_no: 1000
Name: Akshit
Total Balance: 5000

1: Add a new Account
2: Deposit
3: Withdraw
4: Show details
5: Exit
Enter your choice: 3
Enter your account number: 1000
Enter the amount: 1000
Your current balance is: 4000
```

```
1: Add a new Account
2: Deposit
3: Withdraw
4: Show details
5: Exit
Enter your choice: 4
Enter your account number: 1000
Account_no: 1000
Name: Akshit
Total Balance: 4000

1: Add a new Account
2: Deposit
3: Withdraw
4: Show details
5: Exit
Enter your choice: 5
Exiting!!!
```

7. WAP to display taxi details, Customer name and total fare that must be calculated as:
For first 5 km, fare is 50 rs., for next 10 kms, fare is 12 rs./km, for next 15 kms fare is 8 rs./km and 5 rs./km for more than 25 kms.

INPUT:

```
#include<iostream>

using namespace std;

int main()

{

    int fare=0,km,taxi_no;

    string cust_name;

    cout<<"Enter Taxi No: ";

    cin>>taxi_no;

    cout<<"Enter Customer Name: ";

    cin>>cust_name;

    cout<<"Enter how many KM: ";

    cin>>km;

    if(km<=5)
```

```
    {
        fare=50;
    }
    else if(km>5 && km<=15)
    {
        fare=50+(km-5)*12;
    }
    else if(km>15 && km<=30)
    {
        fare=50+120+(km-15)*8;
    }
    else if(km>30)
    {
        fare=50+120+120+(km-30)*5;
    }
    cout<<"\n---BILL---\n";
    cout<<"\nTaxi No is: "<<taxi_no;
    cout<<"\nCustomer Name is: "<<cust_name;
    cout<<"\nYour Fare is: "<<fare;
    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 1\7_Taxi_Fair.exe
Enter Taxi No: 40
Enter Customer Name: Akshit
Enter how many KM: 500

---BILL---

Taxi No is: 40
Customer Name is: Akshit
Your Fare is: 2640
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

8. Write a program to display time in hours, minutes and seconds after adding integer value to the time type object, show the use of constructors.

INPUT:

```
#include<iostream>
#include<conio.h>

using namespace std;

class Time{
    int hours, minutes, seconds;
    public:
    Time()
    {

    }
    Time(int hrs, int min, int sec)
    {
        hours = hrs;
        minutes = min;
        seconds = sec;
    }
    void show()
    {
        cout<<this->hours<<":"<<this->minutes<<":"<<this->seconds<<endl;
    }

};
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
int main()
{
    int h, m, s;
    cout<<"\n Enter Hours input: ";
    cin>>h;
    cout<<"\n Enter Minites input: ";
    cin>>m;
    cout<<"\n Enter Seconds input: ";
    cin>>s;

    Time time(h, m, s);
    cout<<"\n Output is : ";

    time.show();
    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 1\8_Time_Const.exe

Enter Hours input: 7

Enter Minites input: 15

Enter Seconds input: 30

Output is : 7:15:30
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

9. Construct the class named person (Data member: age, name), Write member functions: detail(), display(). Write a program that will accept the detail of four persons and create function such that it will find the max age from the supplied detail.

INPUT:

```
#include<iostream>
using namespace std;
class Person{
public:
	string name;
	float age;
public:
	Person(){


	}
	Person(string n, float g){
		name = n;
		age = g;
	}
	void detail(){
		cout<<"Enter the person name: ";
		cin>>name;
		cout<<"Enter the person age: ";
		cin>>age;
	}
	void display(){
		cout<<"\n\nName is: "<<name;
		cout<<"\nAge is: "<<age;
```

```
	}
	friend void max_age(Person p1, Person p2, Person p3, Person p4);
};

void max_age(Person p1, Person p2, Person p3, Person p4)
{
	if(p1.age > p2.age && p1.age > p3.age && p1.age > p4.age)
	{
		cout<<"\n\nMaximum age is: "<<p1.age;
		cout<<"\nThe name of the person: "<<p1.name;
	}
	else if(p2.age > p1.age && p2.age > p3.age && p2.age > p4.age)
	{
		cout<<"\n\nMaximum age is: "<<p2.age;
		cout<<"\nThe name of the person: "<<p2.name;
	}
	else if(p3.age > p1.age && p3.age > p2.age && p3.age > p4.age)
	{
		cout<<"\n\nMaximum age is: "<<p3.age;
		cout<<"\nThe name of the person: "<<p3.name;
	}
	else
	{
		cout<<"\n\nMaximum age is: "<<p4.age;
		cout<<"\nThe name of the person: "<<p4.name;
	}
}
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
int main(){

	Person t1,t2,t3,t4;


	t1.detail();
	t2.detail();
	t3.detail();
	t4.detail();
	cout<<"\nDisplaying Data:-";
	t1.display();
	t2.display();
	t3.display();
	t4.display();
	max_age(t1, t2, t3, t4);
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 1\9_Class_Person_Display_Detail.exe
Enter the person name: Akshit
Enter the person age: 21
Enter the person name: Sagar
Enter the person age: 22
Enter the person name: Sijo
Enter the person age: 23
Enter the person name: Yash
Enter the person age: 25

Displaying Data:-

Name is: Akshit
Age is: 21

Name is: Sagar
Age is: 22

Name is: Sijo
Age is: 23

Name is: Yash
Age is: 25

Maximum age is: 25
The name of the person: Yash
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

11. Create class DATE having date, month and year as data members. Update the date with user given days using friend function, and display new date.

INPUT:

```
#include <iostream>
using namespace std;
class Date
{
public:
    int date;
    int month;
    int year;
    friend void set(int, Date &);
    void get(Date date)
    {
        cout << "Date:" << date.date << endl;
        cout << "Month:" << date.month << endl;
        cout << "Year:" << date.year;
    }
};
void set(int days, Date &date)
{
    date.year = days / 365;
    date.month = (days % 365) / 30;
    date.date = (days % 365) % 30;
}
int main()
{
    Date date;
    int days;
    cout << "Enter days:";
    cin >> days;
    set(days, date);
    date.get(date);
    return 0;
}  }
    return 0;
}
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 1\11_Date.exe
Enter days: 35
Date: 5
Month: 1
Year: 0
```

12. Create class VECTOR (int x, int y, int z). Using parameterized constructor with default arguments, initialize the data members. Also using member function add two objects of this class and display resultant value using member functions.

INPUT:

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

class VECTOR
{
    int x;
    int y;
    int z;

    public:
    VECTOR(int x=0, int y=0, int z=0)
    {
        this->x = x;
        this->y = y;
        this->z = z;
    }

    void addingvectors(VECTOR other)
    {
        cout<<"X: "<<x + other.x<<endl;
        cout<<"Y: "<<y + other.y<<endl;
        cout<<"Z: "<<z + other.z<<endl;
    }
```

```
};

int main()
{
    VECTOR v1(25);
    VECTOR v2(25,75,100);
    v1.addingvectors(v2);
    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 1\12_Vector.exe
X: 50
Y: 75
Z: 100
```

13. Design an airline reservation data structure that contains the following data:

Flight number
Originating airport code (3 characters)
Destination airport code (3 characters)
Departure time
Arrival time

Write a program that lists all the planes that leave from two airports specified by the user.

INPUT:

```
#include <iostream>

#include <string>

using namespace std;

class Flight

{

private:

    string flightNo;
```

```
    string originated;
    string destination;
    string departure;
    string arrival;


public:
    void set_details(string flightNo, string originated, string destination,
string departure, string arrival)
    {
        this->flightNo = flightNo;
        this->originated = originated;
        this->destination = destination;
        this->departure = departure;
        this->arrival = arrival;
    }
    void get_details()
    {
        cout << "Flight no: " << this->flightNo << endl;
        cout << "Originating Airport Code: " << this->originated << endl;
        cout << "Destination Airport Code: " << this->destination << endl;
        cout << "Departure time: " << this->departure << endl;
        cout << "Arrival time: " << this->arrival << endl;
    }
};


int main()
{
    int n;
```

```
    cout << "Enter number of flights: ";
    cin >> n;
    Flight flights[n];
    for (int i = 0; i < n; i++)
    {
        string flightNo, originated, destination, departure, arrival;
        cout << "\nFlight Number: " << i + 1 << " \n";
        cout << "Enter Flight no: ";
        cin >> flightNo;
        cout << "Enter Originating Airport code: ";
        cin >> originated;
        cout << "Enter Destination Airport code: ";
        cin >> destination;
        cout << "Enter departure time: ";
        cin >> departure;
        cout << "Enter Arrival time: ";
        cin >> arrival;
        flights[i].set_details(flightNo, originated, destination, departure,
arrival);
    }
    for (int i = 0; i < n; i++)
    {
        cout << "\nFlight Number: " << i + 1 << " \n";
        flights[i].get_details();
    }

    return 0;
}
```

OOCP ASSIGNMENT-1

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 1\13_Flight.exe

Flight Number: 1
Enter Flight no: 1
Enter Originating Airport code: 123
Enter Destination Airport code: 321
Enter departure time: 12
Enter Arrival time: 15

Flight Number: 2
Enter Flight no: 2
Enter Originating Airport code: 789
Enter Destination Airport code: 987
Enter departure time: 3
Enter Arrival time: 5

Flight Number: 1
Flight no: 1
Originating Airport Code: 123
Destination Airport Code: 321
Departure time: 12
Arrival time: 15

Flight Number: 2
Flight no: 2
Originating Airport Code: 789
Destination Airport Code: 987
Departure time: 3
Arrival time: 5
```

Rollwala Computer Center

OBJECT ORIENTED CONCEPTS AND PROGRAMMING

Assignment-2

Feb 20, 2022

Name: **Akshit Trivedi**

Roll No: **40**

Course: **Master of Computer Application**

Sem: **1**

OOCP ASSIGNMENT-2

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

PATTERN PROGRAMS

1. Simple Number Triangle Pattern.

INPUT:

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

int main()
{
    int i,j,k,l,n;
    cout<<"Enter the Range=";
    cin>>n;

    for(i=1;i<=n;i++)
    {
        for(k=1;k<=i;k++)
        {
            cout<<i;
        }

    cout<<"\n";
    }

return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 01_simpl
e_pattern.cpp -o 01_simple_pattern } ; if ($?) { .\01_simple_pattern }
Enter the Range=5
1
22
333
4444
55555
```

2. Inverted Pyramid of Numbers.

INPUT:

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

OOCP ASSIGNMENT-2

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
int main()
{
    int rows=6;

    int k=1;

    for(int i = rows; i >= 1; --i)
    {

        for(int j = 0; j < i-1; ++j)
        {
            cout << k<<" ";
        }

        k++;
        cout << endl;
    }
    return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 02_inver
ted_pyramid.cpp -o 02_inverted_pyramid } ; if ($?) { .\02_inverted_pyramid }
1 1 1 1 1
2 2 2 2
3 3 3
4 4
5
```

3. Half Pyramid Pattern of Numbers.

INPUT:

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

int main()
{
    int rows=5;

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

```
            cout<<j<< " ";
        }
        cout <<"\n";
    }
    return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 03_half_
pyramid.cpp -o 03_half_pyramid } ; if ($?) { .\03_half_pyramid }
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

4. Inverted Pyramid of Descending Numbers.

INPUT:

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

int main()
{
    int rows=5;

    for(int i = rows; i >= 1; --i)
    {

        for(int j = 0; j < i; ++j)
        {
            cout << i<<" ";
        }

        cout << endl;
    }

    return 0;
}
```

OOCP ASSIGNMENT-2

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 04_inver
ted_pyramid_of_desending_numbers.cpp -o 04_inverted_pyramid_of_desending_numbers } ; if ($?) { .\04_inverted_pyramid
_of_desending_numbers }
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

5. Inverted Pyramid of the Same Digit.

INPUT:

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

int main()
{
    int rows=5;

    int num = 1;

    for (int i=5; i>=0; i--)
    {
        for(int j=0; j<=i; j++)
            {
                cout<<num;
            }

        printf("\n");

    }
    return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 05_inver
ted_pyramid_of_same_digit.cpp -o 05_inverted_pyramid_of_same_digit } ; if ($?) { .\05_inverted_pyramid_of_same_digit
 }
111111
11111
1111
111
11
1
```

OOCP ASSIGNMENT-2

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

6. Reverse Pyramid of Numbers.

INPUT:

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

int main()
{
    int i,j,k,rows=5,count;

    for(i=1; i<=rows; i++)
    {
        for (int j=i; j>=1; j--)
        {
            cout<<j;
        }
        cout<<"\n";
    }
    return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 06_rever
ce_pyramid.cpp -o 06_reverce_pyramid } ; if ($?) { .\06_reverce_pyramid }
1
21
321
4321
54321
```

7. Inverted Half Pyramid Number Pattern.

INPUT:

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

int main()
 {
    int rows;

    rows = 5;
```

```
    for(int i=5; i>=0; i--)
    {
        for(int j=0; j<=i+1; j++)
        {
            cout<<j;
        }
        cout<<"\n";
    }

    return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 07_inver
ted_half_pyramid.cpp -o 07_inverted_half_pyramid } ; if ($?) { .\07_inverted_half_pyramid }
0123456
012345
01234
0123
012
01
```

8. Pyramid of Natural Numbers Less Than 10.

INPUT:

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

int main()
 {
    int currentNumber = 1, stop = 2, rows=3;

    for(int i=0;i<rows; i++)
    {
        for(int j=1; j<stop; j++)
        {
            cout<<currentNumber <<" ";
            currentNumber++;
        }
        cout<<endl;
        stop +=2;
    }
```

```
    return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 08_pyram
id_of_natural_number.cpp -o 08_pyramid_of_natural_number } ; if ($?) { .\08_pyramid_of_natural_number }
1
2 3 4
5 6 7 8 9
```

9. Reverse Pattern of Digits from 10.

INPUT:

```
#include <iostream>
using namespace std;
int main()
 {
   int start=1, currentNumber , stop = 2, rows;

    currentNumber = stop;

    for(int i=2;i<6; i++)
    {
        for(int j=start; j<stop; j++)
        {
            currentNumber=currentNumber-1;
            cout<<currentNumber <<" ";

        }
        cout<<endl;

        start=stop;

        stop +=i;

        currentNumber=stop;
    }

   return 0;
}
```

Roll No: 40 **Name: Akshit Trivedi**
Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 09_rever
ce_pattarn.cpp -o 09_reverce_pattarn } ; if ($?) { .\09_reverce_pattarn }
1
3 2
6 5 4
10 9 8 7
```

10. Unique Pyramid Pattern of Digits.

INPUT:

```
#include <iostream>
using namespace std;
int main()
 {
   int rows=6;

    for(int i=1; i<=rows; i++)
    {
        for(int j=1; j<i-1; j++)
        {
            cout<<j<<" ";
        }
        for(int j=i-1; j>0; j--)
        {
            cout<<j<<" ";
        }
        cout<<endl;
    }

   return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 10_uniqu
e_pyramid.cpp -o 10_unique_pyramid } ; if ($?) { .\10_unique_pyramid }

1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
```

OOCP ASSIGNMENT-2

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

11. Connected Inverted Pyramid Pattern of Numbers.

INPUT:

```
#include <iostream>
using namespace std;
int main()
 {
    int rows = 6;

    for(int i=0; i<=rows; i++)
    {
        for(int j=rows-1; j>i; j--)
        {
            cout<<j<<"";
        }
        for(int l=0; l<i; l++)
        {
            cout<<"";
        }
        for(int k=i+1; k<rows; k++)
        {
            cout<<k<<"";
        }
        cout<<endl;
    }
    return 0;
 }
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 11_inver
t_pyramid.cpp -o 11_invert_pyramid } ; if ($?) { .\11_invert_pyramid }
5432112345
54322345
543345
5445
55
```

12. Even Number Pyramid Pattern.

INPUT:

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

```
int main()
 {
    int rows = 5;
    int LastEvenNumber = 2 * rows;
    int evenNumber = LastEvenNumber;

    for(int i=1; i<=rows; i++)
    {
        evenNumber = LastEvenNumber;

        for(int j=0; j<i; j++)
        {
            cout<<evenNumber<<" ";

            evenNumber -= 2;
        }
        cout<<endl;
    }
    return 0;
 }
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ tempCode
RunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
10
10 8
10 8 6
10 8 6 4
10 8 6 4 2
```

13. Pyramid of Horizontal Tables.

INPUT:

```
#include<iostream>

using namespace std;
 int main()
 {
    int rows = 6;

    for(int i=0; i<=rows; i++)
    {
```

```
        for(int j=0; j<=i; j++)
        {
            cout<< i*j ;
        }
        cout<<endl;
    }
    return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 13_Horiz
ontal_tables.cpp -o 13_Horizontal_tables } ; if ($?) { .\13_Horizontal_tables }
0
01
024
0369
0481216
0510152025
061218243036
```

14. Pyramid Pattern of Alternate Numbers.

INPUT:

```
#include<iostream>

using namespace std;
 int main()
 {
    int rows = 5;

    for(int i=1; i<=rows; i++)
    {
        for(int j=1; j<=i; j++)
        {
            cout<< i*2-1 ;
        }
        cout<<endl;
    }
    return 0;
 }
```

OOCP ASSIGNMENT-2

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 14_Pyram
id_Alternate.cpp -o 14_Pyramid_Alternate } ; if ($?) { .\14_Pyramid_Alternate }
1
33
555
7777
99999
```

15. Mirrored Pyramid (Right-angled Triangle) Pattern of Numbers.

INPUT:

```
#include<iostream>

using namespace std;
 int main()
 {
    int rows = 5;

    for(int i=1; i<=rows; i++)
    {
        int num=1;
        for(int j=rows; j>0; j--)
        {
            if(j > i)
            {
                cout<<" ";
            }
            else
            {
                cout<<num;
                num+=1;
            }
        }
        cout<<endl;
    }
    return 0;
 }
```

OOCP ASSIGNMENT-2

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 15_Mirro
red_Pyramid.cpp -o 15_Mirrored_Pyramid } ; if ($?) { .\15_Mirrored_Pyramid }
    1
   12
  123
 1234
12345
```

16. Equilateral Triangle with Stars (Asterisk Symbol).

INPUT:

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

int main()
{
    int i, j, rows;

    cout<<"Enter Number of rows : ";
    cin>>rows;

    for(i=1; i<=rows; i++)
    {
        for(j=i; j<rows; j++)
        {
            cout<<" ";
        }

        for(j=1; j<=(2*i-1); j++)
        {
            cout<<"*";
        }

        cout<<"\n";
    }
    return 0;
 }
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ 16_Equil
ateral_Triangle.cpp -o 16_Equilateral_Triangle } ; if ($?) { .\16_Equilateral_Triangle }
Enter Number of rows : 5
    *
   ***
  *****
 *******
*********
```

17. Downward Triangle Pattern of Stars.

INPUT:

```
#include<iostream>
using namespace std;
int main()
{
    int i, space, j,rows;
    cout<<"Enter Number of Rows: ";
    cin>>rows;
    for(i=1; i<=rows; i++)
    {
        for(space=1; space<i; space++)
            cout<<" ";
        for(j=i; j<=rows; j++)
        {
            cout<<"* ";
        }
        cout<<endl;
    }
    cout<<endl;
    return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ tempCode
RunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter Number of Rows: 5
* * * * *
 * * * *
  * * *
   * *
    *
```

OOCP ASSIGNMENT-2

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

18. Pyramid Pattern of Stars.

INPUT:

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

int main()
{
    int space, rows;

    cout <<"Enter number of rows: ";
    cin >> rows;

    for(int i = 1, k = 0; i <= rows; ++i, k = 0)
    {
        for(space = 1; space <= rows-i; ++space)
        {
            cout <<"  ";
        }

        while(k != 2*i-1)
        {
            cout << "* ";
            k++;
        }
        cout << endl;
    }
    return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ tempCode
RunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter number of rows: 5
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

OOCP ASSIGNMENT-2

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

19. Hourglass Pattern Program.

INPUT:

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

void pattern(int rows_no)
{
    int i, j, k;

    for (i = 1; i <= rows_no; i++) {

        for (k = 1; k < i; k++)
            cout << " ";

        for (j = i; j <= rows_no; j++)
            cout << j << " ";

        cout << endl;
    }

    for (i = rows_no - 1; i >= 1; i--) {

        for (k = 1; k < i; k++)
            cout << " ";

        for (j = i; j <= rows_no; j++)
            cout << j << " ";

        cout << endl;
    }
}

int main()
{
    int rows_no;
    cout<<"Enter Number of Rows: ";
    cin>>rows_no;

    pattern(rows_no);
    return 0;
```

```
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ tempCode
RunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter Number of Rows: 5
1 2 3 4 5
 2 3 4 5
  3 4 5
   4 5
    5
   4 5
  3 4 5
 2 3 4 5
1 2 3 4 5
```

20. Pascal's Triangle Program.

INPUT:

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

int main()
{
    int rows, count = 0, count1 = 0, k = 0;

    cout << "Enter number of rows: ";
    cin >> rows;

    for(int i = 1; i <= rows; ++i)
    {
        for(int space = 1; space <= rows-i; ++space)
        {
            cout << "  ";
            count++;
        }

        while(k != 2*i-1)
        {
            if (count <= rows-1)
            {
                cout << i+k << " ";
                count++;
            }
            else
```

OOCP ASSIGNMENT-2

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
            {
                count1++;
                cout << i+k-2*count1 << " ";
            }
            k++;
        }
        count1 = count = k = 0;

        cout << endl;
    }
    return 0;
}
```

OUTPUT:

```
PS C:\MCA\MCA-1\OOCP\Practical Assignment 2> cd "c:\MCA\MCA-1\OOCP\Practical Assignment 2\" ; if ($?) { g++ tempCode
RunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter number of rows: 5
        1
      2 3 2
    3 4 5 4 3
  4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

Rollwala Computer Center

OBJECT ORIENTED CONCEPTS AND PROGRAMMING

Assignment-3

Feb 06, 2022

Name: **Akshit Trivedi**

Roll No: **40**

Course: **Master of Computer Application**

Sem: **1**

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

CALL BY REFERENCE

1. **Write a Program to create class 'coordinate'(int x-point and int y-point) to display coordinate on inverse planes (i.e. convert coordinating points x & y negative) using member function which accept object of class coordinate and display negative values.**

INPUT:

```
#include<iostream>
using namespace std;
class Coordinate
{
    int x,y;
public:
    Coordinate(){}
    Coordinate(int a,int b){
        this->x=a;
        this->y=b;
    }
    void setting_inverse(Coordinate &obj);
};
void Coordinate::setting_inverse(Coordinate &obj)
{
    obj.x=x-(x+x);
    obj.y=y-(y+y);
    cout<<"\nInversing the Values: ";
    cout<<"\nValue of A: "<<obj.x<<endl;
    cout<<"Value of B: "<<obj.y<<endl;
}
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
int main()
{
    int a=0,b=0;
    cout<<"Enter value of A: ";
    cin>>a;
    cout<<"Enter value of B: ";
    cin>>b;

    Coordinate obj(a,b);
    obj.setting_inverse(obj);
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\1_Cordinate.exe
Enter value of A: 5
Enter value of B: 10

Inversing the Values:
Value of A: -5
Value of B: -10
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**
Class: MCA-1 **Year: 2021-22**

CONSTRUCTOR

3. WAP Class NUM array that is initialized with 5 elements by parameter passed from main as an array for an array A[5] as data member and also compute following function on it.

1) Array elements of class with maximum and minimum value.
2) Average / mean value of data members of class.
3) Array elements with sorted values. (Ascending & Descending)
4) Product of Array elements with a scalar variable.
5) Sum of all Array elements.
6) Prime numbers from Array elements.
7) Search particular number from list of Array elements.
8) Factorial of selected element from an array.

Also show use of copy constructor by passing reference of another object of an array.

INPUT:

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

class Num
{
    private:
    int arr[5];
    int i, j, max, min;

    public:
        Num(int a[])
        {
            cout<<"\nArray is : ";
```

```
            for(j=0; j<5; j++)
            {
                arr[j]=a[j];
                cout<<" "<<arr[j];
            }
        }

    // 1. Array elements of class with maximum and minimum value.
    void max_min()
    {
        max = arr[0];
        for (i = 0; i < 5; i++)
        {
            if (max < arr[i])
                {
                    max = arr[i];
                }
        }

        min = arr[0];
        for (i = 0; i < 5; i++)
        {
            if (min > arr[i])
                {
                    min = arr[i];
                }
        }
    cout << "\n Largest element : " << max;
```

```
    cout << "\n Smallest element : " << min;
    }

    // 2) Average / mean value of data members of class.
    void avrage()
    {
        float sum=0.0;
        float avg;

        for (i = 0; i < 5; i++)
        {
            sum = sum + arr[i];
        }

        avg = sum/5;

        cout<<"\n Average of this Elements is : "<<avg;
    }

    // 3) Array elements with sorted values. (Ascending & Descending)
    void sort_fun()
    {
        int i, j, min, temp;

        for (i = 0; i < 4; i++)
        {
            min = i;
            for (j = i + 1; j < 5; j++)
```

```
            {
                 if (arr[j] < arr[min])
                {
                     min = j;
                }
            }
            temp = arr[i];
            arr[i] = arr[min];
            arr[min] = temp;
        }

        cout<<"\n\n Array in Ascending order is : ";
            for(j=0; j<5; j++)
            {
                cout<<" "<<arr[j];
            }

        cout<<"\n Array in Descending order is : ";
            for(j=4; j>=0; j--)
            {
                cout<<" "<<arr[j];
            }
    }

    // 4) Product of Array elements with a scalar variable.
    void product_scal()
    {
        int prod;
```

```
        cout<<"\n Enter Scaler Variable : ";
        cin>>prod;

        cout<<"\t Answer Array is : ";
        for(int i=0; i<5 ; i++)
        {
            arr[i] = arr[i] * prod;
            cout<<arr[i]<<' ';
        }
    }

    // 5) Sum of all Array elements.
    void sum()
    {
        int sum=0;

        for(int i=0; i<5 ; i++)
        {
           sum = sum + arr[i];
        }

        cout<<"\nSum of array is : "<< sum;
    }

    // 6) Prime numbers from Array elements.
    void prime_num()
    {
        int flag, prime[5],size=0;
```

```
        for(i=0;i<5;i++)
        {
          flag=0;
          for(j=2;j<arr[i];j++)
            {
               if(arr[i]%j==0)
               {
                    flag=1;
                    break;
               }
            }
           if(flag==0)
            {
               prime[size]=arr[i];
               size++;
            }
        }
          printf("\n\n Prime Numbers in Above Array : ");
          for(i=0; i<size ;i++)
          {
              cout<<" "<<prime[i];
          }
    }

    // 7) Search particular number from list of Array elements.
    void search()
    {
```

```
        int flag=0,search;

        cout<<"\n Enter Number you Want to find : ";
        cin>>search;

        for (i = 0; i < 5; i++)
        {
            if (arr[i] == search)
            {
                cout<<"\n Yes, This element is present in array.";
                flag=0;
                break;
            }
            else
            {
                flag = 1;
            }
        }
        if(flag==1)
        {
            cout<<"\n No, This element is not present in array";
        }
    }

    // 8) Factorial of selected element from an array.
    void factorial()
    {
        int fact = 1, i, sel, flag=0;
```

```
cout<<"\n\n select any on element of this array : ";
cin>>sel;

 for (i = 0; i < 5; i++)
 {
    if (arr[i] == sel)
    {
        flag=0;
        break;
    }
    else
    {
        flag = 1;
    }
 }
if(flag==1)
 {
    cout<<"\n No, This element is not present in array";
 }
 else if (flag == 0)
 {
    for (i = 2; i <= sel; i++)
    {
        fact =fact * i;
    }
    cout<<"\n Factorial of selected element is: "<<fact;
 }
```

```
    }
};

int main()
{
    int i,A[5];
      cout<<"Enter 5 Elements: ";
    for(i=0; i<5; i++)
    {
        cout<<"\n Enter Element "<<i+1<<": ";
        cin>>A[i];
    }

    Num n(A);

    Num n1(n);

    n1.max_min();
    n1.avrage();
    n1.sort_fun();
    n1.product_scal();
    n1.sum();
    n1.prime_num();
    n1.search();
    n1.factorial();

    return 0;
}
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\3_Constructor.exe
Enter 5 Elements:
 Enter Element 1: 5

 Enter Element 2: 1

 Enter Element 3: 3

 Enter Element 4: 4

 Enter Element 5: 2

Array is :  5 1 3 4 2
 Largest element : 5
 Smallest element : 1
 Average of this Elements is : 3

 Array in Ascending order is :  1 2 3 4 5
 Array in Descending order is :  5 4 3 2 1
 Enter Scaler Variable : 2
         Answer Array is : 2 4 6 8 10
Sum of array is : 30

 Prime Numbers in Above Array :  2
 Enter Number you Want to find : 4

 Yes, This element is present in array.

 select any on element of this array : 6

 Factorial of selected element is: 720
```

4. WAP to show the functionality of copy constructor with class swap(int x, int y) after swapping values of data members of an object which would be passed as an object for initialization of an another object.

INPUT:

```
#include <iostream>

using namespace std;


class Swap
```

```
{
      int a, b;


    public:
    Swap(int a, int b)
    {
        this->a = a;            //40
        this->b = b;            //108
    }
    Swap(Swap &s)
    {
        this->a = s.a;
        this->b = s.b;
    }
    void show()
    {
        cout<<"\n\nValue of a :"<<a<<"\t value of B : "<<b<<endl;
    }
    void swap()
    {
        int temp;


        cout << "\nBefore Swapping: " << a << " " << b;


        temp = a;
        a = b;
        b = temp;
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
        cout << "\nAfter Swapping: " << a << " " << b;      // a=108 b=40
    }
};

int main()
{
      Swap s1(40, 108);
    s1.swap();
    s1.show();

    Swap s2(s1);        //copy call
      s2.show();

      return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\4_Swap_Constructor.exe

Before Swapping: 40 108
After Swapping: 108 40

Value of a :108  value of B : 40


Value of a :108  value of B : 40
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

5. WAP to swap the concept of multiple constructor or constructor overloading with class Faculty (int id, char * name, float salary). Initialize data members using various ways (i.e. default constructor, parameterized constructor, copy constructor) and also show the use of implicit and explicit method for object declaration.

INPUT:

```
#include<iostream>
#include<string.h>

using namespace std;

class Faculty
{
    int id;
    // string name;
    char * name;
    float salary;

    public:
        Faculty()                       //default
        {
           id = 1;
           name= "Sagar";
           salary = 2000;
        }
```

```
        Faculty(int id, char * name, float salary)                    //
perameterzied
        {
           this->id = id;
           this->name= name;
           this->salary = salary;
        }



        Faculty(Faculty &f4)                                          //
Copy Constructor
        {
           id = f4.id;
        // strcpy(name, f1.name);
           name= f4.name;
           salary = f4.salary;
        }


        void display()
        {
           cout<<"\n\n Faculty Information  "<<endl;
           cout<<"=========================================";
           cout<<"\n Id : "<<id;
           cout<<"\n Name : "<<name;
           cout<<"\n Salary : "<<salary<<endl;
        }


};
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
int main()
{
    Faculty f;
    f.display();

    Faculty f1(2,"Sijo",40000);        //  Implicit call
    f1.display();

   //  Faculty f2 = Faculty(3,"Akki",50000);       // Explicit call
   //  f2.display();

    Faculty f4=f1;
    f4.display();

    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\5_Faculty.exe

 Faculty Information
==========================================
 Id : 1
 Name : Yash
 Salary : 1500

 Faculty Information
==========================================
 Id : 2
 Name : Akshit
 Salary : 5000

 Faculty Information
==========================================
 Id : 2
 Name : Akshit
 Salary : 5000
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OPERATOR OVERLOADING

6. WAP to use binary operator + add two object of class Day having day, month and year as its data members and display result in form of date (dd – mm – yyyy).

INPUT:

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


class Day
{
	int days;
	int month;
	int year;
	public:
			Day()
			{
				days = 0;
				year = 0;
				month = 0;
			}


			void setData()
			{
				int temp;
				cout<<"Enter total days: ";
				cin>>temp;
				year = temp / 365;
				temp = temp - year*365;
```

```
                        month = temp / 30;
                        temp = temp - month*30;
                        days = temp;
                }

                void getData()
                {
                        cout<<"Total Days: "<< days<<endl;
                        cout<<"Total Months: "<< month <<endl;
                        cout<<"Total Years: "<< year <<endl;
                }

        Day operator + (Day &other);
        void display()
        {
                cout<<"Date:"<<days<<"-"<<month<<"-"<<year<<endl;
        }
};

 Day Day :: operator + (Day &other)
{
        Day newDay;
        int d, m, y;
        d = days + other.days;
        m = month + other.month;
        y = year + other.year;
        int temp = d / 365;
        y = y + temp;
```

```
      d = d - temp*365;
      temp = d / 30;
      m = m + temp;
      d = d - temp*30;
      temp = m / 12;
      y = y + temp;
      m = m - temp;
    newDay.days = d;
      newDay.month = m;
      newDay.year = y;
      return newDay;
}

int main()
{
    Day d1, d2, d3;
      d1.setData();
      cout<<"d1 Data:"<<endl;
      d1.getData();
      d2.setData();
      cout<<"d2 Data:"<<endl;
      d2.getData();
      d3 = d1 + d2;
      cout<<"d3 Data:"<<endl;
      d3.getData();
    return 0;
}
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**
Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\6_Day_Binary_Operator.exe
Enter total days: 400
d1 Data:
Total Days: 5
Total Months: 1
Total Years: 1
Enter total days: 25
d2 Data:
Total Days: 25
Total Months: 0
Total Years: 0
d3 Data:
Total Days: 0
Total Months: 2
Total Years: 1
```

7. WAP to show use of unary operators (++) and (–). Also perform addition and subtraction of value of data members n1 and n2 of an object of class 'Num' and display the result.

INPUT:

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

class Num {
private:
      int n1,n2,i,ans;

public:

      Num()
```

```
    {
        n1=0;
        n2=0;
    }

    void setData()
    {
        cout<<"Enter N1: ";
        cin>>n1;
        cout<<"Enter N2: ";
        cin>>n2;
    }
    void operator ++()
{
    ++n1;
      ++n2;
}

    void operator --()
    {
        --n1;
        --n2;
    }

    void display()
    {
        cout << "n1: " << n1 << endl;
        cout << "n2: " << n2 << endl;
```

```
	}
};

int main()
{
	Num num1, num2;
	cout<<"Num1:"<<endl;
	num1.setData();
	cout << "Before increment Num1: "<<endl;
	num1.display();

	++num1;

	cout << "After pre increment Num1: "<<endl;
	num1.display();

	cout<<"Num2:"<<endl;
	num2.setData();
	cout << "Before decrement Num2: "<<endl;
	num2.display();

	--num2;

	cout << "After pre decrement Num2: "<<endl;
	num2.display();

	return 0;
}
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**
Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\7_Add_Sub_Unary_Operator.exe
Num1:
Enter N1: 25
Enter N2: 20
Before increment Num1:
n1: 25
n2: 20
After pre increment Num1:
n1: 26
n2: 21
Num2:
Enter N1: 10
Enter N2: 15
Before decrement Num2:
n1: 10
n2: 15
After pre decrement Num2:
n1: 9
n2: 14
```

8. WAP Perform the functionality of insertion (<<) and extractor (>>) operator with object of class time having data members as hr, min, sec (i.e. if object T of class time directly can be passed at insertor and extractor to display it).

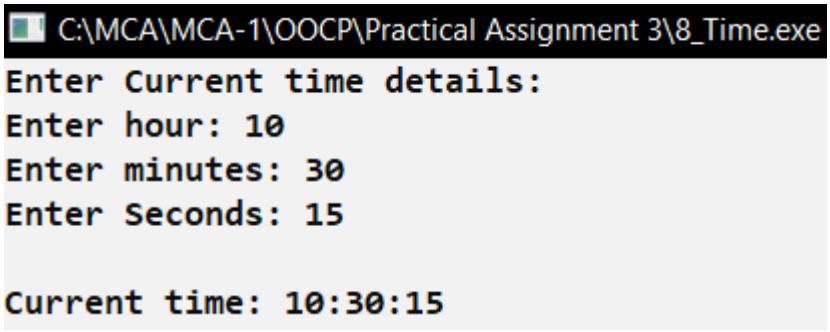
INPUT:

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


class Time
{
```

```
        int hr,min,sec;

        public:

                friend ostream & operator <<(ostream &out,Time t);
                friend istream & operator >>(istream &in,Time &t);
};

ostream & operator << (ostream &out,Time t)
{
        cout<<t.hr<<":"<<t.min<<":"<<t.sec;
        return out;
}

istream & operator >>(istream &in,Time &t)
{
        cout<<"Enter hour: ";
        in>>t.hr;
        cout<<"Enter minutes: ";
        in>>t.min;
        cout<<"Enter Seconds: ";
        in>>t.sec;

        return in;
}

int main()
{
```

```
	Time t;
	cout<<"Enter Current time details: "<<endl;
	cin>>t;


	cout<<endl<<"Current time: "<<t;
	return 0;
}
```

OUTPUT:



OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

INHERITANCE

9. WAP to create base class book having int id and char * name as data members and respective functionality, show following types of inheritance and display the details of each kind of books, also calculate the total no of each type of books in proper format.

- **Simple inheritance with derived class Sales**
- **hierarchical inheritance with derived classes academics and thrillers**
- **Hybrid inheritance with derived classes as above and in addition final derivation of class Sales**

Show use of constructor and destructor in above examples of inheritance.

INPUT:

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


class Book
{
    int id;
    char *name;
    public:
        static int totalBookCount;
        Book()
        {
            totalBookCount++;
            name = (char *)malloc(15);
        }


        ~Book()
        {
```

```
            free(name);
            cout<<"The book memory has been freed"<<endl;
        }

        void set_book()
        {
            cout<<"Enter book id:  ";
            cin>>id;
            cout<<"Enter Book Name:  ";
            cin>>name;
        }
        void get_book()
        {
            cout<<"Book id:  "<<id<<endl;
            cout<<"Book Name:  "<<name<<endl;
            cout<<"Total Books:  "<<totalBookCount<<endl;
        }
};

class Sales : public Book
{
    int salesid;
    string sname;
    public:

    Sales()     //Constructor will call set_sales when the Sales object is
created
    {
```

```
        set_sales();
    }

    ~Sales()
    {
        cout<<"Sales destroyer called"<<endl;
    }
    void set_sales()
    {
        cout<<"Enter salesid:  ";
        cin>>salesid;
        cout<<"Enter SalesPerson Name:  ";
        cin>>sname;
        set_book();
    }

    void get_sales()
    {
        cout<<"\n\nSales Details"<<endl;
        cout<<"Salesid:  "<<salesid<<endl;
        cout<<"Sales Person Name: "<<sname<<endl;
        get_book();
    }
};

class Academics : public Sales
{
    string course;
```

```
    public:

    static int academicsCount;

        Academics()
        {
            academicsCount++;
            cout<<"Enter the course:  ";
            cin>>course;
        }

        void get_academic_details()
        {
            get_sales();
            cout<<"Book type:  Academics"<<endl;
            cout<<"Course:  "<<course<<endl;
            cout<<"Academic book count:  "<<academicsCount<<endl;
        }
};

class Thriller : public Sales
{
    int rating;

    public:
        static int ThrillerCount;

        Thriller()
```

```
        {
            ThrillerCount++;
            cout<<"Enter rating:  ";
            cin>>rating;
        }

        void get_thriller_details()
        {
            get_sales();
            cout<<"Book type:  Thriller"<<endl;
            cout<<"Rating:  "<<rating<<endl;
            cout<<"Thriller book count:  "<<ThrillerCount<<endl;
        }
};

int Book::totalBookCount=0;
int Thriller::ThrillerCount=0;
int Academics::academicsCount=0;

int main()
{
    Academics a1;
    a1.get_academic_details();
    Thriller t1;
    t1.get_thriller_details();

}
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
Enter salesid:  40
Enter SalesPerson Name:  Akshit
Enter book id:  100
Enter Book Name:  OOCP
Enter the course:  MCA


Sales Details
Salesid:  40
Sales Person Name: Akshit
Book id:  100
Book Name:  OOCP
Total Books:  1
Book type:  Academics
Course:  MCA
Academic book count:  1
Enter salesid:  5
Enter SalesPerson Name:  Yash
Enter book id:  200
Enter Book Name:  Sherlock
Enter rating:  5


Sales Details
Salesid:  5
Sales Person Name: Yash
Book id:  200
Book Name:  Sherlock
Total Books:  2
Book type:  Thriller
Rating:  5
Thriller book count:  1
Sales destroyer called
The book memory has been freed
Sales destroyer called
The book memory has been freed
```

10. WAP to create student having data members (rollno, name, stream) as base class. Derive class subject with marks of 5 subjects and apply respective functionality, also read the marks from class arts derived from class subject. Calculate final result and display details of each student from derived class. (multilevel inheritance).

INPUT:

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

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

class Student
{
    protected:
        int rno;
        string name;
        string stream;

    void set_stud()
    {
        cout<<"Enter Roll No.:  ";
        cin>>rno;
        cout<<"Enter Name:  ";
        cin>>name;
    }

    void display_stud()
    {
        cout<<"Rollno:  "<<rno<<endl;
        cout<<"Name:  "<<name<<endl;
        cout<<"Stream:  "<<stream<<endl;
    }

};

class marks : public Student
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
{
    int math;
    int phy;
    int chem;
    int guj;
    int hindi;

    public:
        void set_marks()
        {
            cout<<"Enter marks of Math:  ";
            cin>>math;
            cout<<"Enter marks of Physics:  ";
            cin>>phy;
            cout<<"Enter marks of Chemistry:  ";
            cin>>chem;
            cout<<"Enter marks of Gujarati:  ";
            cin>>guj;
            cout<<"Enter marks of Hindi:  ";
            cin>>hindi;
        }

        void display_marks()
        {
            cout<<"Math:  "<<math<<endl;
            cout<<"Physics:  "<<phy<<endl;
            cout<<"Chemistry:  "<<chem<<endl;
            cout<<"Gujarati:  "<<guj<<endl;
```

```
            cout<<"Hindi:  "<<hindi<<endl;
        }
};

class Arts : public marks
{
    public:
        Arts()
        {
            set_stud();
            set_marks();
        }

        void display_details()
        {
            cout<<"\n\nDetails of Arts Student"<<endl;
            display_stud();
            display_marks();
        }
};

int main()
{
    Arts a1;
    a1.display_details();
    return 1;
}
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**
Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\10_Student.exe
Enter Roll No.:  40
Enter Name:  Akshit
Enter marks of Math:  90
Enter marks of Physics:  95
Enter marks of Chemistry:  92
Enter marks of Gujarati:  93
Enter marks of Hindi:  91


Details of Arts Student
Rollno:  40
Name:  Akshit
Stream:
Math:  90
Physics:  95
Chemistry:  92
Gujarati:  93
Hindi:  91
```

11. WAP to determine class 'ring' from two base class the 'diamond' and 'gold' the details of ring including its price(multiple inheritance).

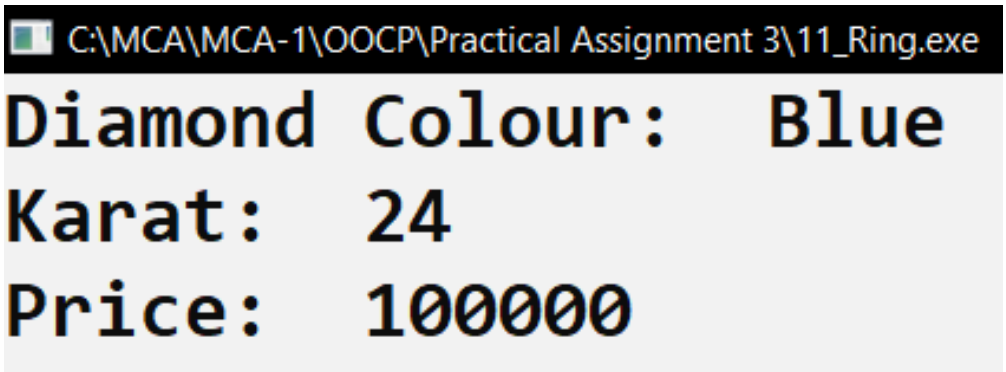
INPUT:

```
#include<iostream>

// #include<cstring>

using namespace std;


class Diamond

{
```

```
    protected:
        string colour;
    public:
        Diamond(string colour)
        {
            this->colour = colour;
        }
};


class Gold
{
    protected:
        int karat;
    public:
        Gold(int karat)
        {
            this->karat = karat;
        }
};


class Ring : public Gold, public Diamond
{


    float price;
    public:
        Ring(float price, int karat, string colour):Gold(karat),
Diamond(colour)
        {
```

```
            this->price = price;
        }


        void display()
        {
            cout<<"Diamond Colour:  "<<colour<<endl;
            cout<<"Karat:  "<<karat<<endl;
            cout<<"Price:  "<<price<<endl;
        }
};


int main()
{
    Ring r1(100000, 24, "Blue");
    r1.display();
    return 1;
}
```

OUTPUT:



OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

POINTER

12. WAP to create class 'hospital' having data as incharge-doctor, license _no and worth in rupees. Take two objects and show the use of pointers to an object.

INPUT:

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

class hospital
{
    string incharge_doctor;
    int license_no;
    float worth;

public:
    /*void input_Details(string doc_name, int lic_no, float wor)
    {
        incharge_doctor=doc_name;
        license_no=lic_no;
        worth=wor;
    }*/
    void input_Details()
    {
        cout<<"Enter Doctor Name: ";
        cin>>incharge_doctor;
        cout<<"Enter License No: ";
        cin>>license_no;
        cout<<"Enter Worth in Rs. : ";
        cin>>worth;
    }
    void Details()
    {
        cout<<"\n\n----Displaying Details----";
        cout<<"\nName of Doctor is: "<<incharge_doctor;
        cout<<"\nLicense No is: "<<license_no;
        cout<<"\nWorth in Rs. is: "<<worth;
    }

};
int main()
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
{
    hospital h1,h2,*hptr;
    hptr=&h1;
    //hptr->input_Details("Akshit",40,5000.75);
    cout<<"\nUsing Object h1 & hptr: \n";
    h1.input_Details();
    hptr->Details();
      //(*hptr).Details();

      hptr=&h2;

    cout<<"\n\nUsing Pointer hptr & h2: \n";
    hptr->input_Details();
    h2.Details();

    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\12_Hospital_Pointer_to_Object.exe

Using Object h1 & hptr:
Enter Doctor Name: Akshit
Enter License No: 40
Enter Worth in Rs. : 100000


----Displaying Details----
Name of Doctor is: Akshit
License No is: 40
Worth in Rs. is: 100000

Using Pointer hptr & h2:
Enter Doctor Name: Yash
Enter License No: 108
Enter Worth in Rs. : 50000


----Displaying Details----
Name of Doctor is: Yash
License No is: 108
Worth in Rs. is: 50000
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

13. Write a program to create 'income' having data members like emp_code and salary. Show the concept of "This pointer", pointer to find out whose income was higher among 3 employees.

INPUT:

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

class income
{
	int emp_code;
	float salary;

public:
	/*void input_details(int id, float sal)
	{
		this->emp_code=id;
		this->salary=sal;
	}*/
	void input_details()
	{
		cout<<"\nEnter Employee Code: ";
		cin>>emp_code;
		cout<<"Enter Salary: ";
		cin>>salary;
	}
	void display_details()
	{
		cout<<"\nEmployee Code is: "<<emp_code;
		cout<<"\nSalary is: "<<salary;
	}
	friend void max_sal(income p1, income p2, income p3);
};

void max_sal(income p1, income p2, income p3)
{
	if(p1.salary > p2.salary && p1.salary > p3.salary)
	{
		cout<<"\nMaximum salary is: "<<p1.salary<<endl;
		cout<<"\nEmployee Code is: "<<p1.emp_code<<endl;
	}
	else if(p2.salary > p1.salary && p2.salary > p3.salary)
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
	{
		cout<<"\nMaximum salary is: "<<p2.salary<<endl;
		cout<<"\nEmployee Code is: "<<p2.emp_code<<endl;
	}
	else
	{
		cout<<"\nMaximum salary is: "<<p3.salary<<endl;
		cout<<"\nEmployee Code is: "<<p3.emp_code<<endl;
	}
}

int main()
{
	income i1,i2,i3;
	/*i1.input_details(40,4000.123);
	i2.input_details(108,8562.123);
	i3.input_details(420,25000.85);*/

	i1.input_details();
	i2.input_details();
	i3.input_details();
	cout<<"\nDisplaying Employee Details: ";
	i1.display_details();
	i2.display_details();
	i3.display_details();
	cout<<"\n\n\nMaximum Salary Employee Details: ";
	max_sal(i1, i2, i3);
	return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\13_Income_This_Pointer.exe

Enter Employee Code: 40
Enter Salary: 10000

Enter Employee Code: 108
Enter Salary: 20000

Enter Employee Code: 5
Enter Salary: 30000

Displaying Employee Details:
Employee Code is: 40
Salary is: 10000
Employee Code is: 108
Salary is: 20000
Employee Code is: 5
Salary is: 30000


Maximum Salary Employee Details:
Maximum salary is: 30000

Employee Code is: 5
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

CONTAINERSHIP

14. WAP to create class DOB with data field day, month, year and show its use in class faculty having data fields name, DOB, salary along with the appropriate functions display the details of 5 faculties and also calculate their age.

INPUT:

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

class Faculty
{
    class DOB
    {
        int day;
        int month;
        int year;
        public:
            void setDOB()
            {
                cout<<"Date of birth:"<<endl;
                cout<<"Enter day: ";
                cin>>day;
                cout<<"Enter month: ";
                cin>>month;
                cout<<"Enter year: ";
                cin>>year;
            }
```

```
        void getDOB()
        {
            cout<<"DOB: "<<day<<"-"<< month <<"-"<<year<<endl;
            calculateAge();
        }

        void calculateAge()
        {
            time_t currentTime = time(0);
            struct tm *myTime = localtime(&currentTime);
            int age;
            int currentYear= myTime->tm_year+1900;

            if(myTime->tm_mon+1 < month)
            {
                age = currentYear - year - 1;
            }
            else
            {
                age = currentYear - year;
            }
            cout<<"Age: "<<age<<endl;
        }


};

string name;
```

```
    float sal;
    DOB dob;

    public:
            void setDetails()
            {
                cout<<"Enter Name: ";
                cin>>name;
                cout<<"Enter Salary: ";
                cin>>sal;
                dob.setDOB();
            }

            void showDetails()
            {
                cout<<"Name: "<<name<<endl;
                cout<<"Salary: "<<sal<<endl;
                dob.getDOB();
            }

            void calculateAge()
            {
                dob.calculateAge();
            }

};

int main()
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
{
    Faculty f[2];
    for(int i=0;i<2;i++)
    {
        f[i].setDetails();
        f[i].showDetails();
    }
    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\14_DOB_Faculty.exe
Enter Name: Akshit
Enter Salary: 250000
Date of birth:
Enter day: 25
Enter month: 12
Enter year: 2000
Name: Akshit
Salary: 250000
DOB: 25-12-2000
Age: 21
Enter Name: Sagar
Enter Salary: 300000
Date of birth:
Enter day: 1
Enter month: 1
Enter year: 2000
Name: Sagar
Salary: 300000
DOB: 1-1-2000
Age: 22
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

NAMESPACE

15. WAP to create namespace out_nmspc to have class counter (int a, int b). Show the use counters in another class by including/using already created namespace.

INPUT:

```
#include <iostream>
using namespace std;
namespace out_nmspc
{
    class counter
    {
        int a, b;
        public:
            void set_details()
            {
                cout<<"Enter a: ";
                cin>>a;
                cout<<"Enter b: ";
                cin>>b;
            }

            void get_details()
            {
                cout<<"a: "<<a<<endl;
                cout<<"b: "<<b<<endl;
            }
    };
}
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
using namespace out_nmspc;

class Holder
{
    counter c;
    int x;
    int y;
    public:
        void set_details()
        {
            c.set_details();
            cout<<"Enter x: ";
            cin>>x;
            cout<<"Enter y: ";
            cin>>y;
        }

        void get_details()
        {
            c.get_details();
            cout<<"x: "<<x<<endl;
            cout<<"y: "<<y<<endl;
        }
};


int main()
```

```
{
    Holder h;
    h.set_details();
    h.get_details();
    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\15_Namespace.exe
Enter a: 50
Enter b: 20
Enter x: 55
Enter y: 88
a: 50
b: 20
x: 55
y: 88
```

16. WAP to create namespace having function for summation (total of values). Show its use in class 'marks' of students, display total marks of subjects using namespace.

INPUT:

```
#include <iostream>
using namespace std;
namespace sum
{
    class Addtion
    {
        int sum;
        public:
            Addtion()
```

```
            {
                sum = 0;
            }
            int add(int *arr, int size)
            {
                sum = 0;
                for (int i = 0; i < size; i++)
                {
                    sum += arr[i];
                }
                return sum;
            }
    };
}
class Mark
{
    int marks[5];
    sum::Addtion obj;
    public:
        void set()
        {
            for (int i = 0; i < 5; i++)
            {
                cout << "Enter marks:";
                cin >> marks[i];
            }
        }
        void getsum()
```

```
        {
            cout << "Sum:" << obj.add(marks, sizeof(marks) /
            sizeof(marks[0]));
        }
};
int main()
{
    Mark obj;
    obj.set();
    obj.getsum();
    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\16_Student_Marks.exe
Enter marks:90
Enter marks:95
Enter marks:91
Enter marks:92
Enter marks:93
Sum:461
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

VIRTUAL FUNCTION

17. Write a program to create class student having virtual function display and show its use with derived class PGDCSA and MCA having display() for the details of all students.

INPUT:

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

class Student
{
    public:
        string name;
    public:
        Student(){ }
        Student(string name)
        {
            this->name = name;
        }

        void set_name()
        {
            cout<<"Enter your name:  ";
            cin>>name;
        }

        virtual void display() = 0;
};
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
class PGDCSA : public Student
{
    int rno;
    string course;
    public:
        PGDCSA(string name, int rno, string course) : Student(name)
        {
            this->rno = rno;
            this->course = course;
        }

        void display()
        {
            cout<<"\nStudent of PGDCA\n\n";
            cout<<"Name:  "<<name<<endl;
            cout<<"Rollno:  "<<rno<<endl;
            cout<<"Course:  "<<course<<endl;
        }
};

class MCA : public Student
{

    int rno;
    int sem;
    public:
        void set_val()
        {
```

```
            set_name();
            cout<<"Enter your Rollno:  ";
            cin>>rno;
            cout<<"Enter your sem:  ";
            cin>>sem;
        }

        void display()
        {
            cout<<"\nStudent of MCA \n\n";
            cout<<"Name:  "<<name<<endl;
            cout<<"Rollno:  "<<rno<<endl;
            cout<<"Semister:  "<<sem<<endl;
        }
};

int main()
{
    PGDCSA p1("Yash" ,108, "MBA");
    MCA m1;
    m1.set_val();
    p1.display();
    m1.display();
    return 0;
}
```

Roll No: 40 Name: Akshit Trivedi
Class: MCA-1 Year: 2021-22

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\17_Virtual_Function_Student.exe
Enter your name:  Akshit
Enter your Rollno:  40
Enter your sem:  1

Student of PGDCA

Name:  Yash
Rollno:  108
Course:  MBA

Student of MCA

Name:  Akshit
Rollno:  40
Semister:  1
```

18. Convert above program to show the concept of pointer to derived class.

INPUT:

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


class Student
{
    public:
        string name;
    public:
        Student(){ }
        Student(string name)
```

```
        {
            this->name = name;
        }

        void set_name()
        {
            cout<<"Enter your name:  ";
            cin>>name;
        }

        virtual void display() = 0;
};

class PGDCSA : public Student
{
    int rno;
    string course;
    public:
        PGDCSA(string name, int rno, string course) : Student(name)
        {
            this->rno = rno;
            this->course = course;
        }

        void display()
        {
            cout<<"\nStudent of PGDCA\n\n";
            cout<<"Name:  "<<name<<endl;
```

```
            cout<<"Rollno:  "<<rno<<endl;
            cout<<"Course:  "<<course<<endl;
        }
};

class MCA : public Student
{
    int rno;
    int sem;
    public:
        void set_val()
        {
            set_name();
            cout<<"Enter your Rollno:  ";
            cin>>rno;
            cout<<"Enter your sem:  ";
            cin>>sem;
        }

        void display()
        {
            cout<<"\nStudent of MCA \n\n";
            cout<<"Name:  "<<name<<endl;
            cout<<"Rollno:  "<<rno<<endl;
            cout<<"Semister:  "<<sem<<endl;
        }

};
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
int main()
{
    PGDCSA p1("Yash" ,108, "MBA");
    MCA m1;
    Student *s1;
    s1 = &m1;
    m1.set_val();
    s1->display();
    s1 = &p1;
    s1->display();
    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\18_Pointer_To_Derived.exe
Enter your name:  Akshit
Enter your Rollno:  40
Enter your sem:  1

Student of MCA

Name:  Akshit
Rollno:  40
Semister:  1

Student of PGDCA

Name:  Yash
Rollno:  108
Course:  MBA
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

TEMPLATES

19. Write a program to generate templates function for swapping values of variables and show its use with integer, float and character type of data as input.

INPUT:

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


template<typename T>
void swapped(T&a , T&b)
{
    T temp;
    temp = a;
    a = b;
    b = temp;
}
int main()
{
    // Integer Swapped
    int i1, i2;
    char c1, c2;
    float f1, f2;
    int op=0;


    while(op!=4)
    {
        cout<<"1: Swap Integer"<<endl;
        cout<<"2: Swap float"<<endl;
```

```
        cout<<"3: Swap Character"<<endl;
        cout<<"4: Exit"<<endl;
        cout<<"Enter your choice: ";
        cin>>op;
        switch(op)
        {
            case 1:
                    cout<<"Enter a(int): ";
                    cin>>i1;
                    cout<<"Enter b(int): ";
                    cin>>i2;
                    cout<<"Integer Swapped:"<<endl;
                    cout<<"a: "<<i1<<endl;
                    cout<<"b: "<<i2<<endl;
                    swapped(i1, i2);
                    cout<<"After Swapping"<<endl;
                    cout<<"a: "<<i1<<endl;
                    cout<<"b: "<<i2<<endl<<endl;
                    break;

            case 2:
                    // Float Swapped
                    float f1, f2;
                    cout<<"Enter f1(float): ";
                    cin>>f1;
                    cout<<"Enter f2(float): ";
                    cin>>f2;
                    cout<<"Float Swapped:"<<endl;
```

```
            cout<<"f1: "<<f1<<endl;
            cout<<"f2: "<<f2<<endl;
            swapped(f1, f2);
            cout<<"After Swapping"<<endl;
            cout<<"f1: "<<f1<<endl;
            cout<<"f2: "<<f2<<endl<<endl;
            break;

    case 3:
            // Character Swapped
            char c1, c2;
            cout<<"Enter c1(character): ";
            cin>>c1;
            cout<<"Enter c2(character): ";
            cin>>c2;
            cout<<"Character Swapped:"<<endl;
            cout<<"c1: "<<c1<<endl;
            cout<<"c2: "<<c2<<endl;
            swapped(c1, c2);
            cout<<"After Swapping"<<endl;
            cout<<"c1: "<<c1<<endl;
            cout<<"c2: "<<c2<<endl;
            break;

    case 4:
            cout<<"Exiting the program!!!";
            break;
```

```
            default:
                    cout<<"Invalid Input";
                    break;
        }
    }
    return 0;
}
```

OUTPUT:

```
Select C:\MCA\MCA-1\OOCP\Practical Assignment 3\19_Template_Swap.exe
1: Swap Integer
2: Swap float
3: Swap Character
4: Exit
Enter your choice: 1
Enter a(int): 5
Enter b(int): 10
Integer Swapped:
a: 5
b: 10
After Swapping
a: 10
b: 5

1: Swap Integer
2: Swap float
3: Swap Character
4: Exit
Enter your choice: 2
Enter f1(float): 5.23
Enter f2(float): 8.63
Float Swapped:
f1: 5.23
f2: 8.63
After Swapping
f1: 8.63
f2: 5.23
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**
Class: MCA-1 **Year: 2021-22**

```
1: Swap Integer
2: Swap float
3: Swap Character
4: Exit
Enter your choice: 3
Enter c1(character): a
Enter c2(character): z
Character Swapped:
c1: a
c2: z
After Swapping
c1: z
c2: a
1: Swap Integer
2: Swap float
3: Swap Character
4: Exit
Enter your choice: 4
Exiting the program!!!
```

20. Write a program to create templates class 'stack' with push function and pop function, show its use by taking number and character type data.

INPUT:

```
#include<iostream>

using namespace std;

#define MAX 10


template <class T>

class Stack
```

```
{
    T stack[MAX];
    int top=-1;
    public:
        void push()
        {
            if(top+1 == MAX)
            {
                cout<<"Stack Overflow"<<endl;
                return;
            }
            cout<<"Enter the element: ";
            cin>>stack[++top];
        }

        void pop()
        {
            if(top==-1)
            {
                cout<<"Stack Underflow"<<endl;
                return;
            }
            cout<<"The element popped is: "<<stack[top--]<<endl;
        }
};
void integerStack()
{
    Stack<int> s1;
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
int op=0;
while(op!=3)
{
    cout<<"1: Integer Stack PUSH"<<endl;
    cout<<"2: Integer Stack POP"<<endl;
    cout<<"3: Exit"<<endl;
    cout<<"Enter your choice: ";
    cin>>op;

    switch(op)
    {
        case 1:
                s1.push();
                break;

        case 2:
                s1.pop();
                break;

        case 3:
                cout<<"Exiting the Integer Stack"<<endl;
                break;

        default:
                cout<<"Invalid Input"<<endl;
                break;
    }
}
```

```
}
void charStack()
{
    Stack<char> s1;
    int op=0;
    while(op!=3)
    {
        cout<<"1: Character Stack PUSH"<<endl;
        cout<<"2: Character Stack POP"<<endl;
        cout<<"3: Exit"<<endl;
        cout<<"Enter your choice: ";
        cin>>op;

        switch(op)
        {
            case 1:
                    s1.push();
                    break;

            case 2:
                    s1.pop();
                    break;

            case 3:
                    cout<<"Exiting the Character Stack"<<endl;
                    break;

            default:
```

```
                    cout<<"Invalid Input";
                    break;
        }
    }
}
int main()
{
    int op1=0;
    while(op1!=3)
    {
        cout<<"1: Integer Stack"<<endl;
        cout<<"2: Character Stack"<<endl;
        cout<<"3: Exit"<<endl;
        cout<<"Enter your choice: ";
        cin>>op1;

        switch(op1)
        {
            case 1:
                    integerStack();
                    break;
            case 2:
                    charStack();
                    break;

            case 3:
                    cout<<"Exiting the program!!!";
                    break;
```

```
            default:
                    cout<<"Invalid Input";
                    break;
        }
    }
    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\20_Stack.exe
1: Integer Stack
2: Character Stack
3: Exit
Enter your choice: 1
1: Integer Stack PUSH
2: Integer Stack POP
3: Exit
Enter your choice: 1
Enter the element: 5
1: Integer Stack PUSH
2: Integer Stack POP
3: Exit
Enter your choice: 1
Enter the element: 10
1: Integer Stack PUSH
2: Integer Stack POP
3: Exit
Enter your choice: 1
Enter the element: 15
1: Integer Stack PUSH
2: Integer Stack POP
3: Exit
Enter your choice: 2
The element popped is: 15
1: Integer Stack PUSH
2: Integer Stack POP
3: Exit
Enter your choice: 2
The element popped is: 10
1: Integer Stack PUSH
2: Integer Stack POP
3: Exit
Enter your choice: 3
Exiting the Integer Stack
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
1: Integer Stack
2: Character Stack
3: Exit
Enter your choice: 2
1: Character Stack PUSH
2: Character Stack POP
3: Exit
Enter your choice: 1
Enter the element: a
1: Character Stack PUSH
2: Character Stack POP
3: Exit
Enter your choice: 1
Enter the element: z
1: Character Stack PUSH
2: Character Stack POP
3: Exit
Enter your choice: 2
The element popped is: z
1: Character Stack PUSH
2: Character Stack POP
3: Exit
Enter your choice: 2
The element popped is: a
1: Character Stack PUSH
2: Character Stack POP
3: Exit
Enter your choice: 3
Exiting the Character Stack
1: Integer Stack
2: Character Stack
3: Exit
Enter your choice: 3
Exiting the program!!!
```

22. Write a program to create templates class 'vehicle' as a base class, also take 'bike' as a derived class(either templates or normal class) show its data in proper formats.

INPUT:

```
#include<iostream>
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
using namespace std;


template <class T>
class Vehicle
{
    T engine_no;
    public:
        void setDetails()
        {
            cout<<"Enter Engine Id: ";
            cin>>engine_no;
        }

        void getDetails()
        {
            cout<<"Engine id: "<<engine_no<<endl;
        }
};

template <class T>
class Bike : Vehicle<T>
{
    string company;
    public:
        void setDetails()
        {
            cout<<"Enter the Company: ";
            cin>>company;
```

```
            Vehicle<T>::setDetails();
        }

        void getDetails()
        {
            cout<<"Company: "<<company<<endl;
            Vehicle<T>::getDetails();
        }
};

int main()
{
    Bike<int> b1;
    b1.setDetails();
    b1.getDetails();
    return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\22_Vehicle.exe
Enter the Company: TVS
Enter Engine Id: 4
Company: TVS
Engine id: 4
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

FILE PROGRAMS

23. **Write a Program to write content in a text file.**
24. **Write a Program to read content from a text file and display it in proper format.**
25. **Write a Program to count number of words from the file.**
26. **Write a Program to count number of lines from the file.**

INPUT:

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

void write_into_file()
{
    string str1;
    ofstream fout;
    fout.open("contents.txt", ios::app);
    cin.ignore();
    cout<<"Enter the contents: ";
    getline(cin, str1);
    fout << str1 + "\n";
    fout.close();
}

void read_from_file()
{
    string str1;
    ifstream fin;
```

```
    fin.open("contents.txt", ios::in);
    while(!fin.eof())
    {
        getline(fin, str1);
        cout<<str1<<endl;
    }
    fin.close();
}

void count_words()
{
    int count=0;
    char ch;
    ifstream fin;
    fin.open("contents.txt", ios::in);
    while(!fin.eof())
    {
        ch = fin.get();
        if(ch==' ' || ch=='\n')
        {
            count++;
        }
    }
    if(ch='\n') count--;
    fin.close();
    cout<<"Number of words: "<<count<<endl;
}
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
void count_lines()
{
    int count=0;
    string str1;
    int lenstr;
    ifstream fin;
    fin.open("contents.txt", ios::in);
    while(!fin.eof())
    {
        count++;
        getline(fin, str1);
    }
    fin.close();
    cout<<"Number of lines: "<<count<<endl;
}

int main()
{
    string str1;
    int op=0;


    while(op!=5)
    {
        cout<<"1: Enter contents into file"<<endl;
        cout<<"2: Read contents from file"<<endl;
        cout<<"3: Count number of words"<<endl;
```

```
        cout<<"4: Count number of lines"<<endl;
        cout<<"5: Exit"<<endl;
        cout<<"Enter your choice: ";
        cin>>op;
        switch(op)
        {
            case 1:
                    write_into_file();
                    break;
            case 2:
                    read_from_file();
                    break;
            case 3:
                    count_words();
                    break;
            case 4:
                    count_lines();
                    break;
            case 5:
                    cout<<"Exiting!!!";
                    break;
        }
    }
    return 0;
}
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\23_24_25_26_File.exe
1: Enter file into file
2: Read file from file
3: Count number of words
4: Count number of lines
5: Exit
Enter your choice: 1
Enter the file: My Name is Akshit.This is OOCP Assignment.
1: Enter file into file
2: Read file from file
3: Count number of words
4: Count number of lines
5: Exit
Enter your choice: 2
My Name is Akshit.This is OOCP Assignment.

1: Enter file into file
2: Read file from file
3: Count number of words
4: Count number of lines
5: Exit
Enter your choice: 3
Number of words: 6
1: Enter file into file
2: Read file from file
3: Count number of words
4: Count number of lines
5: Exit
Enter your choice: 4
Number of lines: 2
```

27. Write a Program to copy contents of file to another file.

INPUT:

```
#include <iostream>

#include <string>

#include <fstream>

using namespace std;
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
int main()
{
string content;
ofstream fout;
ifstream fin;
fout.open("file2.txt");
fin.open("file1.txt", ios::in);
while (!fin.eof())
{
getline(fin, content);
cout << content << endl;
fout << content << endl;
}
fin.close();
fout.close();
return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\27_Copy_Content_File.exe
My Name is Akshit Trivedi.This is OOCP subject Assignemnt.
```

28. Write a program to read array of number and write into a file, also read back from file and display it along with its total.

INPUT:

```
#include <iostream>
#include <string>
#include <fstream>
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
using namespace std;
int main()
{
int arr[5];
for (int i = 0; i < 5; i++)
{
cout << "Enter element:";
cin >> arr[i];
}
ofstream fout;
fout.open("arrayfile.txt");
for (int i = 0; i < 5; i++)
{
fout << arr[i] << endl;
}
fout.close();
ifstream fin;
fin.open("arrayfile.txt", ios::in);
string content;
int sum = 0;
while (!fin.eof())
{
getline(fin, content);
cout << content << endl;
if (content == "")
{
break;
}
```

```
sum += stoi(content);
}
cout << "Total:" << sum;
fin.close();
return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\28_Total_Array_File.exe
Enter element:3
Enter element:2
Enter element:1
Enter element:10
Enter element:20
3
2
1
10
20

Total:36
```

29. Write a program to create a file student to store name and marks of 5 students and then display them.

INPUT:

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
int main()
{
int arr[5];
ofstream fout;
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
fout.open("studentmarks.txt");
for (int i = 0; i < 5; i++)
{
string stu_name;
int stu_marks;
cout << "Enter Student Name:";
cin >> stu_name;
cout << "Enter Student Marks:";
cin >> stu_marks;
fout << stu_name << ":" << stu_marks << endl;
}
fout.close();
ifstream fin;
fin.open("studentmarks.txt", ios::in);
string content;
int sum = 0;
while (!fin.eof())
{
getline(fin, content);
cout << content << endl;
}
fin.close();
return 0;
}
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\29_Student_Marks_File.exe
Enter Student Name:Akshit
Enter Student Marks:90
Enter Student Name:Sijo
Enter Student Marks:95
Enter Student Name:Sagar
Enter Student Marks:98
Enter Student Name:Yash
Enter Student Marks:94
Enter Student Name:Dev
Enter Student Marks:92
Akshit:90
Sijo:95
Sagar:98
Yash:94
Dev:92
```

30. Write a program using file handling technique to create a file data to store Item_code, Item_name and display its price of five items object and display all records with all item details.

INPUT:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main()
{
int arr[5];
ofstream fout;
fout.open("itemfile.txt");
for (int i = 0; i < 5; i++)
{
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
string itemName, itemCode;
int price;
cout << "Enter Item code:";
cin >> itemCode;
cout << "Enter Item name:";
cin >> itemName;
cout << "Enter price:";
cin >> price;
fout << itemCode << " " << itemName << " " << price
<< endl;
}
fout.close();
ifstream fin;
fin.open("itemfile.txt", ios::in);
string content;
int sum = 0;
cout << "code Name price" << endl;
while (!fin.eof())
{
getline(fin, content);
cout << content << endl;
}
fin.close();
return 0;
}
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\30_Item_File.exe
Enter Item code:101
Enter Item name:Redmi
Enter price:5000
Enter Item code:102
Enter Item name:Realme
Enter price:5500
Enter Item code:103
Enter Item name:Apple
Enter price:10000
Enter Item code:105
Enter Item name:Samsung
Enter price:8000
Enter Item code:104
Enter Item name:Vivo
Enter price:7000
code Name price
101 Redmi 5000
102 Realme 5500
103 Apple 10000
105 Samsung 8000
104 Vivo 7000
```

31. Write a program using file handling techniques to create a salary slip for 'n' number of employees by accepting the data (emp_id, emp_name, emp_salary) fields as an input, also display all records of 5 employees and also display net salary by calculating HRA (15% of salary), TA (10% of Salary), DA (20% of Salary), PF (12% of Salary).

INPUT:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
class Employee
```

```
{
string emp_id;
string emp_name;
int emp_salary;
public:
Employee()
{
emp_id = "";
emp_name = "";
emp_salary = 0;
}
void set(string emp_id, string emp_name, int emp_salary)
{
this->emp_id = emp_id;
this->emp_name = emp_name;
this->emp_salary = ((emp_salary)*15 * 10 * 20 * 12)/ 10000;
}
string getName()
{
return emp_name;
}
string getId()
{
return emp_id;
}
int getSalary()
{
return emp_salary;
```

```
}
};
int main()
{
Employee obj[5];
ofstream fout;
fout.open("employeesal.txt");
for (int i = 0; i < 5; i++)
{
string empId, empName;
int empSalary;
cout << "Enter Emp Id:";
cin >> empId;
cout << "Enter Employee name:";
cin >> empName;
cout << "Enter Salary:";
cin >> empSalary;
obj[i].set(empId, empName, empSalary);
fout << obj[i].getId() << " " << obj[i].getName() <<
" " << obj[i].getSalary() << endl;
}
fout.close();
ifstream fin;
fin.open("employeesal.txt", ios::in);
string content;
int sum = 0;
cout << "Id Name salary" << endl;
while (!fin.eof())
```

```
{
getline(fin, content);
cout << content << endl;
}
fin.close();
return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\31_Employee_Salary.exe
Enter Emp Id:40
Enter Employee name:Akshit
Enter Salary:10000
Enter Emp Id:16
Enter Employee name:Sijo
Enter Salary:15000
Enter Emp Id:21
Enter Employee name:Sagar
Enter Salary:11500
Enter Emp Id:108
Enter Employee name:Yash
Enter Salary:3500
Enter Emp Id:11
Enter Employee name:Dev
Enter Salary:8500
Id Name salary
40 Akshit 36000
16 Sijo 54000
21 Sagar 41400
108 Yash 12600
11 Dev 30600
```

32. Write a program to read character from a file after creating a file with a string of country namespace

i) Display it in reverse order

ii) Sort the characters used in name

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

iii) Display the occurrence number of vowels used in country name.

INPUT:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main()
{
string countryNames[100];
ofstream fout;
fout.open("countryname.txt");
for (int i = 0; i < 5; i++)
{
string countryName;
cout << "Enter country name:";
cin >> countryName;
fout << countryName << endl;
}
fout.close();
ifstream fin;
fin.open("countryname.txt", ios::in);
string content;
int i = 0;
while (!fin.eof())
{
getline(fin, content);
// cout << content << endl;
```

```
countryNames[i] = content;

i++;

}

fin.close();

fin.open("countryname.txt");

char ch;

int noOfVowel = 0;

while (!fin.eof())

{

fin.get(ch);

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o'|| ch == 'u' || ch == 'A'
|| ch == 'E' || ch == 'I' || ch =='O' || ch == 'U')

{

noOfVowel++;

}

}

fin.close();

cout << "=======Display in reverse order=======\n";

for (int j = i; j >= 0; j--)

{

cout << countryNames[j] << endl;

}

cout << "=======Sorted order=======\n";

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

{

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

{

if (countryNames[k] > countryNames[k + 1])
```

OOCP ASSIGNMENT-3

Roll No: 40 **Name: Akshit Trivedi**

Class: MCA-1 **Year: 2021-22**

```
{
string temp;
temp = countryNames[k + 1];
countryNames[k + 1] = countryNames[k];
countryNames[k] = temp;
}
}
}
for (int j = 0; j <= i; j++)
{
cout << countryNames[j] << endl;
}
cout << "\n=========================\n";
cout << "Number of vowels:" << noOfVowel;
return 0;
}
```

OUTPUT:

```
C:\MCA\MCA-1\OOCP\Practical Assignment 3\32_Country_Namespace.exe
Enter country name:India
Enter country name:US
Enter country name:UK
Enter country name:Canada
Enter country name:Zimbabve

=======Display in reverse order=======


Zimbabve
Canada
UK
US
India
=======Sorted order=======

Canada
India
UK
US
Zimbabve


=========================
Number of vowels:11
```