SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

1. **An electricity board charges the following rates to user. For the first 100 units → 60p per unit. For the next 200 units→80p per unit. Beyond 300 units→90p per unit. All users are charged a minimum of Rs. 50; if the total amount is more than 300 then an additional surcharges of 15% is added. Write a program to accept name of user consumed and print charges with the its rates.**

Program :

```
#include<iostream>

using namespace std;

class bord_charge{
    public:
    float charge;

    void charges(int unit){

        // Charge Calculation by given rate
        if (unit <= 100 ){
            charge = unit * 0.60;
        }
        else if (unit <= 300)
        {
            charge = (100 * 0.60) + (unit - 100) * 0.80;
        }
        else{
            charge = (100 * 0.60) + (200 * 0.80) + (unit - 300) * 0.90;
        }

        //Minimum charge
        if(charge < 50){
            charge = 50;
        }

        //surcharge calculation
        if(charge > 300){
            charge += charge * 0.15;
        }
```

```cpp

    }

    void display(string name){
        cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"<<endl;
        cout<<"The Consumer Name is : "<<name<<endl;
        cout<<"The Charge Is : "<<charge;
    }

};


int main()
{
    string name;
    int unit;

    bord_charge c1;

    cout<<"Enter the Consumer Name : ";
    cin>>name;

    cout<<"Enter the Amout of unit : ";
    cin>>unit;

    c1.charges(unit);
    c1.display(name);

    return 0;
}
```

Output :

```
PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 01_electricity.cpp -o 01_electricity } ; if ($?) { .\01_electricity }
Enter the Consumer Name : sunil
Enter the Amout of unit : 300
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
The Consumer Name is : sunil
The Charge Is : 220
PS D:\OOPS_ jouranl> 
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

2. **Define a class to represent a bank account. Include the following members: a. Name of the depositor b. Account number c. Type of Account d. Balance amount in the Account Member Functions: a. To assign initial values. b. To deposit an amount. c. To withdraw an amount after checking the balance. d. To display name and balance. Write main program and handle accounts of 5 customers.**

Program :

```
#include<iostream>

using namespace std;

class Bank{

    private:
    string name;
    int ac_number;
    string ac_type;
    double balance;

    public:
    Bank(){}
    void details(string name,int ac_num,string ac_type,double balance = 0)
    {

        this->name = name;
        this->ac_number = ac_number;
        this->ac_type = ac_type;
        this->balance = balance;
    }

    void Deposite(double amount)
    {
        if(amount>0){
            balance+=amount;
            cout<< amount << " Is Succesfully Deposited!!"<<endl;
        }
        else{
            cout<< amount <<"IS not Valid Amount!!"<<endl;
        }
    }
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    void Withdraw(double amount)
    {
        if(amount>0){
            if (amount <= balance)
            {
                balance-=amount;
                cout<< amount <<" IS succesfully Withdraw!!"<<endl;
            }
            else{
                cout<<"Insufficient balance !!"<<endl;
            }
        }
        else{
            cout<< amount <<"IS not Valid Amount!!"<<endl;
        }
    }

    void Display() /*const */{
        cout<<"Account Holder Name : "<< name <<endl;
        cout<<"Balance Is : "<< balance <<endl;
        cout<<"_________________________________"<<endl;
    }

};

int main(){

    Bank accounts[5];

    accounts[0].details("Jeemy",1234,"Saving",3000);
    accounts[1].details("Dency",5678,"Saving",90000);
    accounts[2].details("Robin",1122,"Saving",60000);
    accounts[3].details("Ruby",5729,"Saving",30000);
    accounts[4].details("Deny",9988,"Saving",7000);

    accounts[0].Deposite(7000);
    accounts[1].Withdraw(4000);
    accounts[3].Display();

    for (int i = 0; i <5; i++)
    {
        accounts[i].Display();
```

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
        cout<<"_____________________________________"<<endl;
    }


    return 0;
}
```

Output :

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 02_bank.cpp -o 02_bank } ; if ($?) { .\02_bank }
7000 Is Succesfully Deposited!!
4000 IS succesfully Withdraw!!
Account Holder Name : Ruby
Balance Is : 30000
_____________________________________
Account Holder Name : Jeemy
Balance Is : 10000
_____________________________________
_____________________________________
Account Holder Name : Dency
Balance Is : 86000
_____________________________________
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

_____________________________________
Account Holder Name : Robin
Balance Is : 60000
_____________________________________
_____________________________________
Account Holder Name : Ruby
Balance Is : 30000
_____________________________________
_____________________________________
Account Holder Name : Deny
Balance Is : 7000
_____________________________________
_____________________________________
```

3. **Program to create a class person having members name and age. Derive a class student having member percentage. Derive another class teacher having member salary. Write necessary member function to initialize, read and write data. Also write the main function.**

Program:

```
#include<iostream>

using namespace std;
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
class person{
    public:
    string name;
    int age;
    void getdata()
    {
        cout<<"Enter Name : ";
        cin>>name;

        cout<<"Enter Age : ";
        cin>>age;
    }

    void display()
    {
        cout<<"Student Name : "<< name<<endl;
        cout<<"Student Age : "<<age<<endl;
    }
};

class student : public person{

    public:
    float percentage;

    void getdata()
    {
        person::getdata();
        cout<<"Enter the Percentage : ";
        cin>>percentage;
    }

    void display()
    {
        person::display();
        cout<<"Percentage : "<<percentage<<"%"<<endl;
    }

};

class Teacher : public person{
    public:
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    float salary;

    void getdata()
    {
        person::getdata();
        cout<<"Enter Salary : ";
        cin>>salary;
    }

    void display()
    {
        person::display();
        cout<<"Salary : $"<<salary<<endl;
    }
};

int main()
{
    student s;
    Teacher t;

    cout<<"Enter Student Information : "<<endl;
    cout<<"-----------------------------"<<endl;
    s.getdata();

    cout<<"\n Enter Teacher Information : "<<endl;
    cout<<"-----------------------------"<<endl;
    t.getdata();

    cout<<"\n Student Information : "<<endl;
    cout<<"-----------------------------"<<endl;
    s.display();

    cout<<"\n Teacher Information : "<<endl;
    cout<<"-----------------------------"<<endl;
    t.display();

    return 0;
}
```

Output :

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 03_person.cpp -o 03_person } ; if ($?) { .\03_person }
Enter Student Information :
-----------------------------
Enter Name : sunil
Enter Age : 19
Enter the Percentage : 90

 Enter Teacher Information :
-----------------------------
Enter Name : Chirag
Enter Age : 30
Enter Salary : 30000

 Student Information :
-----------------------------
Student Name : sunil
Student Age : 19
Percentage : 90%

 Teacher Information :
-----------------------------
Student Name : Chirag
Student Age : 30
Salary : $30000
PS D:\OOPS_ jouranl>
```

4. Program to create a class name student having date member name, no & three marks. Write a member function to input name, rollno & marks& calculate percentage.

Program :

```
#include<iostream>

using namespace std;

class student{
    private:
    string name;
    int roll_no;
    float m1,m2,m3;
    float percentage;

    public:
    void getdata()
    {
        cout<<"Enter Student Data :"<<endl;
        cout<<"---------------------";
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
        cout<<"\n Enter Student Name : ";
        cin>>name;
        cout<<"\n Enter Student Roll No: : ";
        cin>>roll_no;
        cout<<"\n Enter Subject 1 Mark : ";
        cin>>m1;
        cout<<"\n Enter Subject 2 Mark : ";
        cin>>m2;
        cout<<"\n Enter Subject 3 Mark: ";
        cin>>m3;

        calculate();
    }

    void calculate()
    {
        float total = m1 + m2 + m3;
        percentage = (total *100) / 300;
    }

    void display()
    {
        cout<<"\n Student Details : "<<endl;
        cout<<"--------------------\n";

        cout<<"Name : "<<name<<endl;
        cout<<"Roll No : "<<roll_no<<endl;
        cout<<"Subject 1 : "<<m1<<endl;
        cout<<"Subject 2 : "<<m2<<endl;
        cout<<"Subject 3 : "<<m3<<endl;
        cout<<"Percentage : "<<percentage<<"%"<<endl;
    }
};

int main()
{
    student s;
    s.getdata();
    s.display();

    return 0;
}
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

Output :

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                                   Code

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 04_student.cpp -o 04_student } ; if ($?) { .\04_student }
Enter Student Data :
--------------------
 Enter Student Name : sunil

 Enter Student Roll No: : 03

 Enter Subject 1 Mark : 90

 Enter Subject 2 Mark : 90

 Enter Subject 3 Mark: 90

 Student Details :
--------------------
Name : sunil
Roll No : 3
Subject 1 : 90
Subject 2 : 90
Subject 3 : 90
Percentage : 90%
PS D:\OOPS_ jouranl>
```

5. **Create one class time which has hour, minute and second as data member. Now write input function to input class values and find time in the form of minute.**

Program :

```cpp
#include<iostream>

using namespace std;

class Time{
    private:
    int hour;
    int minute;
    int second;

    public:
    void getdata()
    {
        cout<<"Enter the Hours : ";
        cin>>hour;

        cout<<"Enter the Minutes : ";
        cin>>minute;
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
        cout<<"Enter the Seconds : ";
        cin>>second;

    }

    int CalculateMinte()
    {
        double Total;
        Total = (hour * 60) + minute + (second / 60);
        return Total;
    }

    void display()
    {
        cout<< hour << " Hours , "<< minute << " Minutes , "<<second << "
Seconds"<<endl;
        cout<<"Total Minutes : "<<CalculateMinte();
    }
};

int main()
{
    Time T1;
    T1.getdata();
    T1.display();

    return 0;
}
```

Output :

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                    Code

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 05_time.cpp -o 05_time } ; if ($?) { .\05_time }
Enter the Hours : 3
Enter the Minutes : 30
Enter the Seconds : 300
3 Hours , 30 Minutes , 300 Seconds
Total Minutes : 215
PS D:\OOPS_ jouranl>
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

6. **Create a class called "Vehicle" which contains data members registration number and fuel type Make getdata() function to input data value. Create class "two-Wheeler "from vehicle which contains data member's distance and mileage Make getdata() function to input data. Use overloading techniques for getdata()function and display the information with fuel used.**

Program :

```
#include<iostream>

using namespace std;

class Vehicle{
    int Re_num;
    string fuel_type;

    public:
    void getdata()
    {
        cout<<" Enter Registration Number : ";
        cin>>Re_num;

        cout<<"Enter Fuel Type (Petrol ,Diesel): ";
        cin>>fuel_type;
    }

    void display()
    {
        cout<<"Registration Number : "<<Re_num<<endl;
        cout<<"Fuel Type : "<<fuel_type<<endl;
    }

};

class Two_wheeler : public Vehicle{
    float Distance , mileage;

    public:
       void getdata()
    {
        Vehicle::getdata();
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
        cout<<"Enter Distance (Km): ";
        cin>>Distance;

        cout<<"Enter Mileage (Km in 1 liter) : ";
        cin>>mileage;
    }

    void display()
    {

        cout<<"--------------------"<<endl;
        cout<<"Vehicle Detais : "<<endl;
        cout<<"--------------------"<<endl;
        Vehicle::display();
        cout<<"Distance : "<<Distance<<" Km"<<endl;
        cout<<"Mileage : "<<mileage<<" Km"<<endl;
        cout<<"Fuel Used : "<<(Distance / mileage) << " Liter"<<endl;
    }
};

int main()
{
    Two_wheeler bike;
    bike.getdata();
    bike.display();

    return 0;
}
```

Output :

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                    Code

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 06_vehicle.cpp -o 06_vehicle } ; if ($?) { .\06_vehicle }
 Enter Registration Number : 1000023567
Enter Fuel Type (Petrol ,Diesel): Diesel
Enter Distance (Km): 40
Enter Mileage (Km in 1 liter) : 70
--------------------
Vehicle Detais :
--------------------
Registration Number : 1000023567
Fuel Type : Diesel
Distance : 40 Km
Mileage : 70 Km
Fuel Used : 0.571429 Liter
PS D:\OOPS_ jouranl>
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

7. **Write a program that consists of two classes: Time12 and Time24. • Time12: This class maintains time in a 12-hour format (e.g., 3:45 PM). • Time24: This class maintains time in a 24-hour format (e.g., 15:45). The program should allow: 1. Conversion between the two time formats (Time12 to Time24 and vice versa). 2. Display of the time in both formats. 3. Input of time in either format and updating it accordingly. 4. Any additional functionality to manipulate or compare time objects if required. Make sure to implement the necessary methods to achieve these functionalities in both classes.**

Program :

```
#include <iostream>
#include <iomanip>
#include <string>

using namespace std;

// Forward declaration of Time24 class
class Time24;

class Time12 {
private:
    int hours;     // 1 to 12
    int minutes;   // 0 to 59
    bool isPM;     // true if PM, false if AM

public:
    // Constructor to initialize time
    Time12(int h = 12, int m = 0, bool pm = false) : hours(h), minutes(m),
isPM(pm) {}

    // Method to input time in 12-hour format
    void inputTime() {
        char meridian[3];
        cout << "Enter time in 12-hour format (hh:mm AM/PM): ";
        scanf("%d:%d %s", &hours, &minutes, meridian);
        isPM = (string(meridian) == "PM" || string(meridian) == "pm");
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    }

    // Method to display time in 12-hour format
    void displayTime() const {
        cout << setfill('0') << setw(2) << hours << ":"
        << setfill('0') << setw(2) << minutes
        << (isPM ? " PM" : " AM") << endl;
    }

    // Method to convert 12-hour time to 24-hour time
    Time24 toTime24() const;

    // Getter for comparison purposes
    int getHours() const { return hours; }
    int getMinutes() const { return minutes; }
    bool getIsPM() const { return isPM; }
};

class Time24 {
private:
    int hours;     // 0 to 23
    int minutes;   // 0 to 59

public:
    // Constructor to initialize time
    Time24(int h = 0, int m = 0) : hours(h), minutes(m) {}

    // Method to input time in 24-hour format
    void inputTime() {
        cout << "Enter time in 24-hour format (hh:mm): ";
        scanf("%d:%d", &hours, &minutes);
    }

    // Method to display time in 24-hour format
    void displayTime() const {
        cout << setfill('0') << setw(2) << hours << ":"
             << setfill('0') << setw(2) << minutes << endl;
    }

    // Method to convert 24-hour time to 12-hour time
    Time12 toTime12() const;

    // Getter for comparison purposes
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    int getHours() const { return hours; }
    int getMinutes() const { return minutes; }
};

// Conversion method from Time12 to Time24
Time24 Time12::toTime24() const {
    int h = hours;
    if (isPM && h != 12) {
        h += 12; // Convert PM to 24-hour time
    } else if (!isPM && h == 12) {
        h = 0;    // Convert midnight case
    }
    return Time24(h, minutes);
}

// Conversion method from Time24 to Time12
Time12 Time24::toTime12() const {
    int h = hours;
    bool pm = false;
    if (h == 0) {
        h = 12; // Midnight case
        pm = false;
    } else if (h == 12) {
        pm = true; // Noon case
    } else if (h > 12) {
        h -= 12;
        pm = true; // PM case
    } else {
        pm = false; // AM case
    }
    return Time12(h, minutes, pm);
}

// Main function
int main() {
    // Create Time12 and Time24 objects
    Time12 t12;
    Time24 t24;

    // Input and display 12-hour format time
    t12.inputTime();
    cout << "Time in 12-hour format: ";
    t12.displayTime();
```

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    // Convert to 24-hour format and display
    t24 = t12.toTime24();
    cout << "Converted to 24-hour format: ";
    t24.displayTime();

    // Input and display 24-hour format time
    t24.inputTime();
    cout << "Time in 24-hour format: ";
    t24.displayTime();

    // Convert to 12-hour format and display
    t12 = t24.toTime12();
    cout << "Converted to 12-hour format: ";
    t12.displayTime();

    return 0;
}
```

Output :

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 07_time24.cpp -o 07_time24 } ; if ($?) { .\07_time24 }
Enter time in 12-hour format (hh:mm AM/PM): 9 40 PM
Time in 12-hour format: 09:00 AM
Converted to 24-hour format: 09:00
Enter time in 24-hour format (hh:mm): Time in 24-hour format: 40:00
Converted to 12-hour format: 28:00 PM
PS D:\OOPS_ jouranl>
```

8. Create two classes DM and DB which store the values of distance. DM stores distance in meters and centimeters. DB stores distances in feet and inches. Writea program that can read values for the class object and add one object of DM with another object of DB. Use a friend function to carry out the addition Operation and this function will display answer in meter and centimeters.

Program :

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

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

class DB;

class DM {
private:
    int meters;
    int centimeters;

public:

    DM(int m = 0, int cm = 0) : meters(m), centimeters(cm) {}

    void input() {
        cout << "Enter distance in meters and centimeters (separated by
space): ";
        cin >> meters >> centimeters;
    }

    // Friend function to add DM and DB distances

    friend DM addDistance(const DM &d1, const DB &d2);

    void display() const {
        cout << "Distance: " << meters << " meters and " << centimeters << "
centimeters." << endl;
    }
};

class DB {
private:
    int feet;
    int inches;

public:

    DB(int ft = 0, int in = 0) : feet(ft), inches(in) {}

    void input() {
        cout << "Enter distance in feet and inches (separated by space): ";
        cin >> feet >> inches;
    }
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    // Friend function to add DM and DB distances

    friend DM addDistance(const DM &d1, const DB &d2);
};

// Friend function to add distances from DM and DB

DM addDistance(const DM &d1, const DB &d2) {

    // Conversion constants

    const float inchesToCm = 2.54;   // 1 inch = 2.54 cm
    const float feetToInches = 12;   // 1 foot = 12 inches

    // Convert DB (feet and inches) to centimeters

    float totalInches = d2.feet * feetToInches + d2.inches;
    float totalCentimetersFromDB = totalInches * inchesToCm;

    // Convert centimeters to meters and centimeters

    int metersFromDB = totalCentimetersFromDB / 100;
    int centimetersFromDB = (int)totalCentimetersFromDB % 100;

    // Add DM's meters and centimeters

    int totalMeters = d1.meters + metersFromDB;
    int totalCentimeters = d1.centimeters + centimetersFromDB;

    // Handle overflow if centimeters are >= 100

    if (totalCentimeters >= 100) {
        totalMeters += totalCentimeters / 100;
        totalCentimeters %= 100;
    }

    // Return the sum as a DM object

    return DM(totalMeters, totalCentimeters);
}

int main() {
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    DM distanceInMeters;
    DB distanceInFeet;

    distanceInMeters.input();
    distanceInFeet.input();

    // Add distances using the friend function

    DM totalDistance = addDistance(distanceInMeters, distanceInFeet);

    cout << "Total distance after addition: "<<endl;
    totalDistance.display();

    return 0;
}
```

Output :

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                    Code

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 08_distance.cpp -o 08_distance } ; if ($?) { .\08_distance }
Enter distance in meters and centimeters (separated by space): 4 45
Enter distance in feet and inches (separated by space): 3 67
Total distance after addition:
Distance: 7 meters and 6 centimeters.
PS D:\OOPS_ jouranl>
```

9. **Write a program to maintain a telephone directory use add() and show() Methods to add new entries and display the telephone numbers of a person when the name of the person is given.**

Program :

```
#include<iostream>

using namespace std;

class telephone{
    private:
        string n;
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
        string number;

    public:

    void add()
    {
        cout<<"Enter the name : ";
        cin>>n;

        cout<<"Enter Phone number : ";
        cin>>number;

        cout << "Entry added successfully." << endl;
    }

    void display()
    {
        cout<<"Name is : "<<n<<endl;
        cout<<"Phone Number : "<<number<<endl;
    }

    void exit()
    {
        cout<<"Exiting....."<<endl;
    }

};

int main()
{

    int choice;
    telephone td;

    do{
        cout<<"Directry Menu : "<<endl;
        cout<<"1.New Entry : "<<endl;
        cout<<"2.Show Details : "<<endl;
        cout<<"3.Exite : "<<endl;
        cout<<"Enter your choice : ";
        cin>>choice;

        switch(choice)
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
        {
            case 1:
                td.add();
                break;

            case 2:
                td.display();
                break;

            case 3:
                td.exit();
                break;

            default :
                cout<<"Invalid Choice !!"<<endl;
                break;
        }

    }while(choice !=3);

    return 0;
}
```

Output :

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                    Code

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 09_Telephonr.cpp -o 09_Telephonr } ; if ($?) { .\09_Telephonr }
PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 09_Telephonr.cpp -o 09_Telephonr } ; if ($?) { .\09_Telephonr }
Directry Menu :
1.New Entry :
2.Show Details :
3.Exite :
Enter your choice : 1
Enter the name : Sunil
Enter Phone number : 928xxxxxxx
Entry added successfully.
Directry Menu :
1.New Entry :
2.Show Details :
3.Exite :
Enter your choice : 2
Name is : Sunil
Phone Number : 928xxxxxxx
Directry Menu :
1.New Entry :
2.Show Details :
3.Exite :
Enter your choice : 3
Exiting.....
PS D:\OOPS_ jouranl>
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

10.Create a base class named Shape to store a double-type value that can be used to compare areas. Derive two specific classes, Triangle and Rectangle, from the base class Shape. Include a member function, getData, in the base class to initialize the base data members, and another function, displayArea, to display the area.

Program :

```
#include<iostream>

using namespace std;

class shape{
    protected:
    double base,height;

    public:
    void getdata(double d1,double d2)
    {
        base = d1;
        height = d2;
    }

    virtual void display(){}
};

class Triangle : public shape{


    public:
    void display() override
    {
        double area = 0.5 * base * height;
        cout<<"Area of Triangle : "<<area<<endl;
    }
};

class Rectangle : public shape{

    public:
```

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    void display() override{
        double area = base * height;
        cout<<"Area of Rectangle : "<<area<<endl;
    }
};

int main()
{
    Triangle triangle;
    Rectangle rectangle;

    triangle.getdata(8.0,3.0);
    triangle.display();

    rectangle.getdata(10.0,5.0);
    rectangle.display();

    return 0;
}
```

OutPut :

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                    Code

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 10_shape.cpp -o 10_shape } ; if ($?) { .\10_shape }
Area of Triangle : 12
Area of Rectangle : 50
PS D:\OOPS_ jouranl>
```

11.Write a program for Tower of Hanoi

Program :

```
#include<iostream>

using namespace std;

class Tower_of_hanoi{

    public:
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    void HanoiTower(int n , string src , string helper , string dest)
    {
        if(n==1)
        {
            cout<<"Transfer Disk " << n << " From " << src << " to " <<
dest<<endl;
            return;
        }

        HanoiTower(n-1,src,dest,helper);

        cout<<"Transfer Disk " << n << " From " << src << " to " <<
dest<<endl;

        HanoiTower(n-1,helper,src,dest);

    }

};

int main()
{
    int n;
    Tower_of_hanoi hanoi;

    cout<<"Enter the Disk Number : ";
    cin>>n;

    // S for source , H for Helper and D for destination
    hanoi.HanoiTower(n,"S","H","D");

    return 0;
}
```

OutPut :

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 11_Tower_of_Hanoi.cpp -o 11_Tower_of_Hanoi } ; if ($?) { .\11_Tower_of_Hano
i }
Enter the Disk Number : 4
Transfer Disk 1 From S to H
Transfer Disk 2 From S to D
Transfer Disk 1 From H to D
Transfer Disk 3 From S to H
Transfer Disk 1 From D to S
Transfer Disk 2 From D to H
Transfer Disk 1 From S to H
Transfer Disk 4 From S to D
Transfer Disk 1 From H to D
Transfer Disk 2 From H to S
Transfer Disk 1 From D to S
Transfer Disk 3 From H to D
Transfer Disk 1 From S to H
Transfer Disk 2 From S to D
Transfer Disk 1 From H to D
PS D:\OOPS_ jouranl>
```

12.Write Program to implement Stack Operations like PUSH, POP, PEEP, UPDATE and DISPLAY using class and object.

Program :

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

#define MAX 5

class Stack {
private:
    int arr[MAX];
    int top;

public:

    Stack() {
        top = -1; // Initially stack is empty
    }

    void push(int value) {
        if (top == MAX - 1) {
            cout << "Stack Overflow! Cannot push " << value << endl;
        } else {
            arr[++top] = value;
            cout << "Pushed " << value << " into the stack." << endl;
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
        }
    }

    void pop() {
        if (top == -1) {
            cout << "Stack Underflow! No element to pop." << endl;
        } else {
            cout << "Popped " << arr[top--] << " from the stack." << endl;
        }
    }

    void peep() {
        if (top == -1) {
            cout << "Stack is empty." << endl;
        } else {
            cout << "Top element is: " << arr[top] << endl;
        }
    }

    void update(int position, int value) {
        if (position > top + 1 || position < 1) {
            cout << "Invalid position! Cannot update." << endl;
        } else {
            arr[position - 1] = value;
            cout << "Updated position " << position << " with value " << value
<< "." << endl;
        }
    }

    void display() {
        if (top == -1) {
            cout << "Stack is empty." << endl;
        } else {
            cout << "Stack elements: ";
            for (int i = 0; i <= top; i++) {
                cout << arr[i] << " ";
            }
            cout << endl;
        }
    }
};

int main() {
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    Stack stack;
    int choice, value, position;

    do {
        cout << "\nStack Operations: ";
        cout << "\n1. PUSH\n2. POP\n3. PEEP\n4. UPDATE\n5. DISPLAY\n6. EXIT";
        cout << "\nEnter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1: // PUSH
                cout << "Enter value to push: ";
                cin >> value;
                stack.push(value);
                break;

            case 2: // POP
                stack.pop();
                break;

            case 3: // PEEP
                stack.peep();
                break;

            case 4: // UPDATE
                cout << "Enter position to update: ";
                cin >> position;
                cout << "Enter new value: ";
                cin >> value;
                stack.update(position, value);
                break;

            case 5: // DISPLAY
                stack.display();
                break;

            case 6: // EXIT
                cout << "Exiting..." << endl;
                break;

            default:
                cout << "Invalid choice! Please try again." << endl;
        }
```

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    } while (choice != 6);

    return 0;
}
```

OutPut :

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

1. PUSH
2. POP
3. PEEP
4. UPDATE
5. DISPLAY
6. EXIT
Enter your choice: 2
Popped 2 from the stack.

Stack Operations:
1. PUSH
2. POP
3. PEEP
4. UPDATE
5. DISPLAY
6. EXIT
Enter your choice: 5
Stack is empty.

Stack Operations:
1. PUSH
2. POP
3. PEEP
4. UPDATE
5. DISPLAY
6. EXIT
Enter your choice:
```

13.Write Program to convert Infix to Postfix Expression using class and object.

Program :

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

class InfixToPostfix {
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
private:
    string infix;
    string postfix;

    // Helper function to get precedence of operators

    int precedence(char op) {
        if (op == '+' || op == '-') {
            return 1; // Precedence of + and - is 1
        } else if (op == '*' || op == '/') {
            return 2; // Precedence of * and / is 2
        } else if (op == '^') {
            return 3; // Precedence of ^ is 3
        }
        return 0;
    }

    // Helper function to check if the character is an operator

    bool isOperator(char c) {
        return (c == '+' || c == '-' || c == '*' || c == '/' || c == '^');
    }

    // Helper function to check if the character is an operand

    bool isOperand(char c) {
        return (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') || (c >= '0'
&& c <= '9');
    }

    // Helper function to check associativity (Right to Left for '^', Left to
Right for others)

    bool isRightAssociative(char op) {
        return (op == '^');
    }

public:

    // Constructor to initialize infix expression

    InfixToPostfix(string expr) : infix(expr), postfix("") {}
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    // Method to convert infix to postfix

    string convert() {

        stack<char> st; // Stack to hold operators and parentheses

        for (char &ch : infix) {

            // If the character is an operand, append it to postfix

            if (isOperand(ch)) {
                postfix += ch;
            }

            // If the character is '(', push it to stack

            else if (ch == '(') {
                st.push(ch);
            }

            // If the character is ')', pop and output from the stack until
'(' is found

            else if (ch == ')') {
                while (!st.empty() && st.top() != '(') {
                    postfix += st.top();
                    st.pop();
                }
                st.pop(); // Remove '(' from the stack
            }

            // If the character is an operator

            else if (isOperator(ch)) {
                while (!st.empty() && precedence(st.top()) >= precedence(ch))
{
                    if (ch == '^' && precedence(st.top()) == precedence(ch)) {
                        break; // Right-associative operators should not pop
from stack
                    }
                    postfix += st.top();
                    st.pop();
                }
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
                st.push(ch); // Push the current operator to stack
            }
        }

        // Pop all remaining operators from the stack

        while (!st.empty()) {
            postfix += st.top();
            st.pop();
        }

        return postfix;
    }
};

int main() {
    string infixExpression;

    // Input infix expression

    cout << "Enter an infix expression: ";
    cin >> infixExpression;

    // Create object of InfixToPostfix class and convert expression

    InfixToPostfix converter(infixExpression);
    string postfixExpression = converter.convert();

    // Display the result

    cout << "Postfix expression: " << postfixExpression << endl;

    return 0;
}
```

OutPut :

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                    Code

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 13_infix_to_postfix.cpp -o 13_infix_to_postfix } ; if ($?) { .\13_infix_to_
postfix }
Enter an infix expression: (A-B/C)*(A/K-L)
Postfix expression: ABC/-AK/L-*
PS D:\OOPS_ jouranl>
```

14.Write Program to convert Infix to Prefix Expression using class and object.

Program :

```
#include <iostream>
#include <stack>
#include <algorithm> // For reverse function
#include <string>
using namespace std;

class InfixToPrefix {
    private:
    string infix;
    string prefix;

    // Helper function to get precedence of operators

    int precedence(char op) {
        if (op == '+' || op == '-') {
            return 1; // Precedence of + and - is 1
        } else if (op == '*' || op == '/') {
            return 2; // Precedence of * and / is 2
        } else if (op == '^') {
            return 3; // Precedence of ^ is 3
        }
        return 0;
    }

    // Helper function to check if the character is an operator

    bool isOperator(char c) {
        return (c == '+' || c == '-' || c == '*' || c == '/' || c == '^');
    }
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
    // Helper function to check if the character is an operand

    bool isOperand(char c) {
        return (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') || (c >= '0'
&& c <= '9');
    }

    // Helper function to check associativity (Right to Left for '^', Left to
Right for others)

    bool isRightAssociative(char op) {
        return (op == '^');
    }

    // Function to reverse the string

    string reverseString(string str) {
        reverse(str.begin(), str.end());
        return str;
    }

    public:
    // Constructor to initialize infix expression

    InfixToPrefix(string expr) : infix(expr), prefix("") {}

    // Method to convert infix to prefix

    string convert() {
        stack<char> st;

        // Reverse the infix expression

        infix = reverseString(infix);

        // Loop through the reversed infix expression

        for (char &ch : infix) {
            // If the character is an operand, append it to prefix
            if (isOperand(ch)) {
                prefix += ch;
            }
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
            // If the character is ')', push it to stack (since we reversed,
')' becomes '('

            else if (ch == ')') {
                st.push(ch);
            }

            // If the character is '(', pop and output from the stack until
')' is found

            else if (ch == '(') {
                while (!st.empty() && st.top() != ')') {
                    prefix += st.top();
                    st.pop();
                }
                st.pop(); // Remove ')' from the stack
            }
            // If the character is an operator

            else if (isOperator(ch)) {
                while (!st.empty() && precedence(st.top()) > precedence(ch)) {
                    prefix += st.top();
                    st.pop();
                }
                st.push(ch); // Push the current operator to stack
            }
        }

        // Pop all remaining operators from the stack

        while (!st.empty()) {
            prefix += st.top();
            st.pop();
        }

        // Reverse the result to get the final prefix expression

        prefix = reverseString(prefix);

        return prefix;
    }
};
```

SDJ INTERNATIONAL COLLEGE

Name : Agarwal Yugen
COURSE : SYBCA
DIVISION : A
ROLL NO : 003

```
int main() {
    string infixExpression;

    // Input infix expression

    cout << "Enter an infix expression: ";
    cin >> infixExpression;

    // Create object of InfixToPrefix class and convert expression

    InfixToPrefix converter(infixExpression);
    string prefixExpression = converter.convert();

    // Display the result

    cout << "Prefix expression: " << prefixExpression << endl;

    return 0;
}
```

OutPut :

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS                    Code

PS D:\OOPS_ jouranl> cd "d:\OOPS_ jouranl\" ; if ($?) { g++ 14_infix_to_prefix.cpp -o 14_infix_to_prefix } ; if ($?) { .\14_infix_to_pr
efix }
Enter an infix expression: (A-B/C)*(A/K-L)
Prefix expression: *-A/BC-/AKL
PS D:\OOPS_ jouranl>
```