**ASSIGNMENT - 2**

Name: **HITARTH GANATRA**

Sem: **II**

Div.: **1**

ID: **22BCA064**

Subject: **Fundamentals of Object Oriented Programming Language**

Q1

**CODE:**

// WAP to print your name, age city and pin code using CLASS

#include<iostream>

using namespace std;

class Info

{

    public:

        string name;

        int age;

        string city;

        int pincode;

        int displayData(){

            cout << "Name : " << name << endl;

            cout << "Age : " << age << endl;

            cout << "City : " << city << endl;

            cout << "Pincode : " << pincode << endl;

        }

};

int main(){

    Info Hitarth;

    Hitarth.name = "Hitarth Ganatra";

    Hitarth.age = 18;

    Hitarth.city = "Rajkot";

    Hitarth.pincode = 360005;

    Hitarth.displayData();

    return 0;

}

Q2

**CODE:**

// WAP to print the area of rectangle by creating class named 'Area' having two functions. First function named as 'setDim' takes the length and breadth of rectangle as parameters and the second function named as 'getArea' returns the area of rectangle. Length and Breadth of the rectangle are entered through keyboard.

#include<iostream>

using namespace std;

class Area{

    private:

        int length, breadth;

    public:

        void setDim(int l, int b){

            length = l;

            breadth = b;

        }

        int getArea(){

            return length \* breadth;

        }

};

int main(){

    Area a;

    int l, b;

    cout << "Enter the length and breadth of the rectangle: ";

    cin >> l >> b;

    a.setDim(l, b);

    cout << "Area of the rectangle is: " << a.getArea() << endl;

    return 0;

}

Q3

**CODE:**

// WAP to display addition, subtraction, multiplication and division of two integers on screen

// Declare Class Calculation

// Declare data member num1 and num2 in private section

// Write member function for initialize num1 and num2

// Write member function for each operation

#include<iostream>

using namespace std;

class Calculation{

    private:

        int num1, num2;

    public:

        int getData(int n1, int n2);

        int add();

        int sub();

        int mul();

        float div();

};

int Calculation::getData(int n1, int n2){

    num1 = n1;

    num2 = n2;

}

int Calculation::add(){

    return num1 + num2;

}

int Calculation::sub(){

    return num1 - num2;

}

int Calculation::mul(){

    return num1 \* num2;

}

float Calculation::div(){

    float result;

    result = (float)num1/num2;

    return result;

}

int main(){

    Calculation q1;

    int n1, n2;

    cout << "Enter two numbers: ";

    cin >> n1 >> n2;

    q1.getData(n1, n2);

    cout << "Addition: " << q1.add() << endl;

    cout << "Subtraction: " << q1.sub() << endl;

    cout << "Multiplication: " << q1.mul() << endl;

    cout << "Division: " << q1.div() << endl;

    return 0;

}

Q4

**CODE:**

// WAP to find area of circle using class

#include<iostream>

using namespace std;

#define PI 3.14

class Area{

    private:

        float radius;

    public:

        Area(float r){

            radius = r;

        }

        float getArea(){

            return PI \* radius \* radius;

        }

};

int main(){

    float r;

    cout << "Enter radius of circle: ";

    cin >> r;

    Area a(r);

    cout << "Area of circle is: " << a.getArea() << endl;

    return 0;

}

Q5

**CODE:**

// WAP to crate a class for student to get and print details of a student i.e. Student ID, Name, Semester, Branch

#include<iostream>

using namespace std;

class Student{

    private:

        string studentID, name, branch;

        int semester;

    public:

        int setData(string i1, string i2, int i3, string i4);

        int displayData();

};

int Student::setData(string i1, string i2, int i3, string i4){

    studentID = i1;

    name = i2;

    semester = i3;

    branch = i4;

}

int Student::displayData(){

    cout << "Student ID : " << studentID << endl;

    cout << "Name : " << name << endl;

    cout << "Semester : " << semester << endl;

    cout << "Branch : " << branch << endl << endl;

}

int main(){

    Student cypher, reyna, killjoy;

    cypher.setData("22BCA064", "Cypher", 2, "BCA");

    reyna.setData("22BCA080", "Reyna", 2, "BCA");

    killjoy.setData("22BCA117", "Killjoy", 2, "BCA");

    cypher.displayData();

    reyna.displayData();

    killjoy.displayData();

    return 0;

}

Q6

**CODE:**

#include<iostream>

using namespace std;

class ATM{

    private:

        int accountNo, balance;

    public:

        int setData(int i1, int i2);

        int withdraw(int amt);

        int deposit(int amt);

};

int ATM::setData(int i1, int i2){

    accountNo = i1;

    balance = i2;

}

int ATM::withdraw(int amt){

    if(amt > balance){

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

    }

    else{

        cout << "The amount " << amt << " has been deducted from your account." << endl;

        balance -= amt;

        cout << "Remaining balance : " << balance << endl << endl;

    }

}

int ATM::deposit(int amt){

    balance += amt;

    cout << "The amount " << amt << " has been deposited to your account." << endl;

    cout << "Updated balance : " << balance << endl << endl;

}

int main(){

    ATM user1, user2, user3;

    user1.setData(123456789, 4000);

    user1.withdraw(2000);

    user1.deposit(1000);

    user2.setData(987654321, 5000);

    user2.withdraw(5500);

    user2.deposit(500);

    user3.setData(123456789, 10000);

    user3.withdraw(10000);

    user3.deposit(500);

    return 0;

}

Q7

**CODE:**

#include<iostream>

using namespace std;

class Student{

    private:

        string studentID, name, branch;

        int sub1, sub2, sub3, sub4, sub5;

        float percentage;

        string studentClass;

    public:

        int setData(string i1, string i2, string i3, int i4, int i5, int i6, int i7, int i8);

        int displayData();

        int calculatePercentage();

        int calculateClass();

};

int Student::setData(string i1, string i2, string i3, int i4, int i5, int i6, int i7, int i8){

    studentID = i1;

    name = i2;

    branch = i3;

    sub1 = i4;

    sub2 = i5;

    sub3 = i6;

    sub4 = i7;

    sub5 = i8;

}

int Student::calculatePercentage(){

    percentage = (sub1 + sub2 + sub3 + sub4 + sub5) / 5.0;

}

int Student::calculateClass(){

    if(percentage >= 80){

        studentClass = "Distinction";

    }

    else if(percentage >= 60){

        studentClass = "First Class";

    }

    else if(percentage >= 40){

        studentClass = "Second Class";

    }

    else{

        studentClass = "Pass";

    }

}

int Student::displayData(){

    cout << "Student ID: " << studentID << endl;

    cout << "Name: " << name << endl;

    cout << "Branch: " << branch << endl;

    cout << "Percentage: " << percentage << endl;

    cout << "Class: " << studentClass << endl << endl;

}

int main(){

    Student chamber, omen, neon;

    chamber.setData("22BCA064", "Chamber", "BCA", 90, 90, 90, 90, 90);

    omen.setData("22BCA080", "Omen", "BCA", 97, 95, 83, 99, 90);

    neon.setData("22BCA117", "Neon", "BCA", 70, 70, 70, 70, 70);

    chamber.calculatePercentage();

    chamber.calculateClass();

    omen.calculatePercentage();

    omen.calculateClass();

    neon.calculatePercentage();

    neon.calculateClass();

    chamber.displayData();

    omen.displayData();

    neon.displayData();

    return 0;

}

Q8

**CODE:**

#include<iostream>

using namespace std;

class Employee{

    private:

        int empcode, salary, income\_tax, netsalary;

    public:

        int da = 0, hra = 0, ta = 500;

        int setData(int code, int sal){

            empcode = code;

            salary = sal;

        }

        int calculations();

        int calculateTax();

        int netSalary();

        int displayData();

};

int Employee::calculations(){

    da = salary + (salary \* 0.74);

    hra = (salary \* 0.1);

}

int Employee::calculateTax(){

    calculations();

    if (salary > 50000)

    {

        income\_tax = salary \* 0.5;

    }

    else{

        income\_tax = 0;

    }

}

int Employee::netSalary(){

    calculateTax();

    netsalary = (salary + da + hra + ta) - income\_tax;

}

int Employee::displayData(){

    netSalary();

    cout << "Employee Code : " << empcode << endl;

    cout << "Salary : " << salary << endl;

    cout << "Income Tax : " << income\_tax << endl;

    cout << "Net Salary : " << netsalary << endl << endl;

}

int main(){

    Employee e1;

    e1.setData(2209, 65000);

    e1.displayData();

    return 0;

}

Q9

**CODE:**

#include<iostream>

using namespace std;

class Employee{

    private:

        int empcode, salary, income\_tax, netsalary;

    public:

        int da = 0, hra = 0, ta = 500;

        int setData(int code, int sal){

            empcode = code;

            salary = sal;

        }

        int calculations();

        int calculateTax();

        int netSalary();

        int displayData();

};

int Employee::calculations(){

    da = salary + (salary \* 0.74);

    hra = (salary \* 0.1);

}

int Employee::calculateTax(){

    calculations();

    if (salary > 50000)

    {

        income\_tax = salary \* 0.05;

    }

    else{

        income\_tax = 0;

    }

}

int Employee::netSalary(){

    calculateTax();

    netsalary = (salary + da + hra + ta) - income\_tax;

}

int Employee::displayData(){

    netSalary();

    cout << "Employee Code : " << empcode << endl;

    cout << "Salary : " << salary << endl;

    cout << "Income Tax : " << income\_tax << endl;

    cout << "Net Salary : " << netsalary << endl << endl;

}

int main(){

    Employee managers[5];

    for (int i = 0; i < 5; i++)

    {

        managers[i].setData(i+1, ((i+1)\*55000));

        managers[i].displayData();

    }

    return 0;

}

Q10

**CODE:**

// WAP to read time in hh:mm:ss and display answer only in seconds

// For e.g :- 2:15:30 then it should display 8130 seconds

#include<iostream>

using namespace std;

class secondsConverter{

    private:

        string time;

        int total\_seconds;

    public:

        int hours, minutes, seconds;

        int setData(int hh, int mm, int ss);

        int convertData();

        int displayData();

};

int secondsConverter::setData(int hh, int mm, int ss){

    hours = hh;

    minutes = mm;

    seconds = ss;

}

int secondsConverter::convertData(){

    hours = hours \* 60 \* 60;

    minutes = minutes \* 60;

    total\_seconds = hours + minutes + seconds;

}

int secondsConverter::displayData(){

    convertData();

    cout << (hours/60)/60 << ":" << (minutes/60)/60 << ":" << seconds << endl;

    cout << "Total seconds : " << total\_seconds << endl << endl;

}

int main(){

    secondsConverter sample1, sample2;

    sample1.setData(2, 15, 30);

    sample1.displayData();

    sample2.setData(2, 15, 31);

    sample2.displayData();

    return 0;

}