**CPP Lab 4**

Roll no. 248121

PRN: 240841220016

Name: Akshay Chavan

1. Create Date class with members day,month ,year.

Write no argument and parameterised constructor .Create two object s and initialize them using no argument and parameterised constructor

respectively.Print date using display function.

**code:**

**Date.cpp**

#include <iostream>

using namespace std;

class Date

{

    int day, month, year;

    //  Default

public:

    Date()

    {

        day = 1;

        month = 1;

        year = 2000;

    }

    //  Parametrized constr

    Date(int day, int month, int year)

    {

        this->day = day;

        this->month = month;

        this->year = year;

    }

    void display()

    {

        cout << "----Display date--------" << endl;

        cout << "Date is " << day << "-" << month << "-" << year << endl;

    }

};

int main()

{

    cout << "----Constructors----" << endl;

    cout << "Using default constr" << endl;

    Date s1;

    s1.display();

    cout << "Using parameterized constr" << endl;

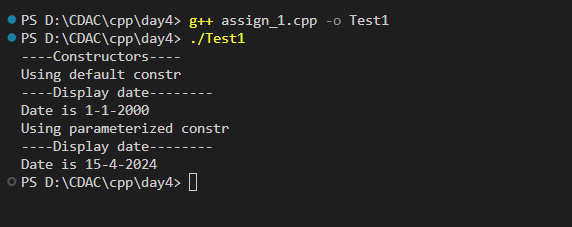
    Date s2(15, 4, 2024);

    s2.display();

    return 0;

}

**output:**

****

1. Create Employee class with members id(int),name(string),dob(Date).Use above created Date class.

Write default and parameterised constructor in Employee Class.Write accept() function to accept information and display() to display emp information.

**code:**

**assign\_2.cpp**

#include <iostream>

using namespace std;

// Date class - named as Date2

class Date2

{

    int day, month, year;

    //  Default

public:

    Date2()

    {

        day = 1;

        month = 1;

        year = 1111;

    }

    //  Parametrized constr

    Date2(int day, int month, int year)

    {

        this->day = day;

        this->month = month;

        this->year = year;

    }

    //  setters

    void setDay(int day)

    {

        this->day = day;

    }

    void setMonth(int month)

    {

        this->month = month;

    }

    void setYear(int year)

    {

        this->year = year;

    }

    void dateDisplay()

    {

        cout << "Date is " << day << "-" << month << "-" << year << endl;

    }

};

class Employee

{

    int id;

    string name;

    Date2 dob;

    static int idGenerator;

public:

    //  defauylt constr

    Employee()

    {

        cout << "------Default constr of Employee called------" << endl;

        id = ++idGenerator;

        name = "John Doe";

        dob = Date2();

    }

    Employee(string name, int day, int month, int year)

    {

        cout << "------Parameterized constr of Employee called------" << endl;

        this->id = ++idGenerator;

        this->name = name;

        this->dob = Date2(day, month, year);

    }

    //      setters

    void setDate(int i, int j, int k)

    {

        dob.setDay(i);

        dob.setMonth(j);

        dob.setYear(k);

    }

    // made this fxn static to be able to get called without object

    static Employee accept()

    {

        string name;

        cout << "Enter name of employee: ";

        cin >> name;

        cout << "Enter date of birth as dd mm yyyy: ";

        int dd, mm, yy;

        cin >> dd >> mm >> yy;

        Employee emp(name, dd, mm, yy);

        return emp;

    }

    //  display

    void display()

    {

        cout << "----Display Employee--------" << endl;

        cout << "Employee: \nid: " << id << endl;

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

        dob.dateDisplay();

    }

};

int Employee::idGenerator = 1000;

int main()

{

    cout << "----Employee creation----" << endl;

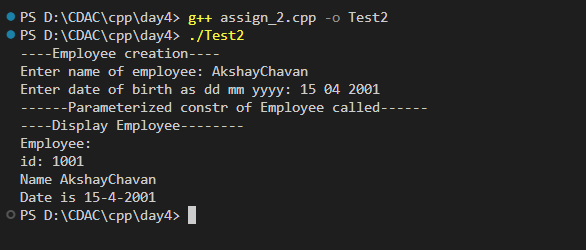
    Employee myEmp = Employee::accept(); // accept info

    myEmp.display();                     // display info

    return 0;

}

**output:**

****

1. Consider that payroll software needs to be developed for computerization of

operations of an ABC organization. The organization has employees.

3.1. Construct a class Employee with following members using private access specifies:

Employee Id integer

Employee Name string

Basic Salary double

HRA double

Medical double=1000

PF double

PT double

Net Salary double

Gross Salary double

Please use following expressions for calculations://Note:Don't accept HRA,PF PT from user

\* HRA = 50% of Basic Salary

\* PF = 12% of Basic Salary

\* PT = Rs. 200

3.2. Write methods to display the details of an employee and calculate the gross

and net salary.

\* Goss Salary = Basic Salary + HRA + Medical

\* Net Salary = Gross Salary – (PT + PF)

Create Object of employee class and assign values and display Details.

**code:**

**assign\_3.cpp**

#include <iostream>

using namespace std;

class Employee2

{

    static int idGenerator;

    int id;

    string name;

    double basicSalary, HRA, medical = 2000, PF, PT, netSalary, grossSalary;

public:

    //  Constrs

    Employee2()

    {

        cout << "------Default of Employee called------" << endl;

        id = ++idGenerator;

        name = "John Doe";

        basicSalary = 0;

    }

    Employee2(string name, double basicSalary)

    {

        this->id = ++idGenerator;

        this->name = name;

        this->basicSalary = basicSalary;

        this->HRA = (basicSalary / 2);

        this->PT = (basicSalary \* 12) / 100;

        this->PF = 200;

        this->grossSalary = basicSalary + this->HRA + this->medical;

        this->netSalary = this->grossSalary - this->PT - this->PF;

        cout << "Employee created using parameterized constr!" << endl;

    }

    double calculateGrossSal()

    {

        return this->grossSalary;

    }

    double calculateNetSal()

    {

        return this->netSalary;

    }

    //  display

    void display()

    {

        cout << "----Display Employee--------" << endl;

        cout << "Employee: " << id << endl;

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

    }

};

int Employee2 ::idGenerator = 1000;

int main()

{

    double salary;

    string name;

    cout << "Enter name: ";

    cin >> name;

    cout << "Enter Salary: ";

    cin >> salary;

    Employee2 emp(name, salary);

    emp.display();

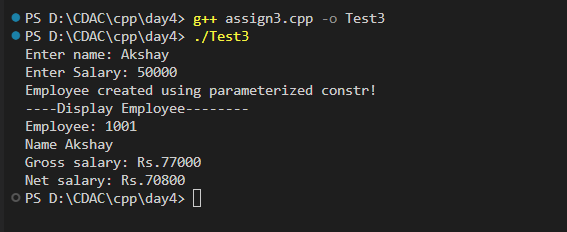
    cout << "Gross salary: Rs." << emp.calculateGrossSal() << endl;

    cout << "Net salary: Rs." << emp.calculateNetSal() << endl;

    return 0;

}

**output:**

****