

Topic: WAP to create a class Account

PIONEER®

Page: 23

Date: / /

i) deposit

ii) display balance

iii) withdrawl

iv) check balance :

```
# include <iostream >
```

```
using name space std;
```

```
class Account
```

```
{
```

```
    char name[100];
```

~~int~~ a cno;

```
protected : int balance;
```

```
public :
```

```
Account()
```

```
{
```

```
cout << "Enter Name" << endl;
```

```
cin >> name;
```

```
cout << "Enter Account number" << endl;
```

```
cin >> aec no;
```

~~cout << "Enter Balance" << endl;~~

```
cin >> balance;
```

```
}
```

```
void display()
```

```
cout << "Name=" << name << endl;
```

```
cout << "Account number = " << aec no << endl;
```

```
}
```

```
};
```

```
class creAcc : public Account
```

{

```
    int pen, dep, with, bal, balW, flag;
```

```
public :
```

```
creAcc () : Account ()
```

{

```
cout << "Enter amount to be deposit" << endl;
```

```
cin >> dep;
```

```
cout << "Enter amount to be withdraw" << endl;
```

```
cin >> with;
```

}

```
void add ()
```

{

```
bal = balance + dep;
```

```
if (with > balance)
```

{

```
flag = 1;
```

}

```
else {
```

```
balW = bal - with;
```

}

```
void pPenalty ()
```

{

```
if (bal < 10000)
```

```
pen = bal - (bal * 0.1);
```

```
3
void display(display B())
{
    display A()
    cout << "Total balance = " << bal << endl;
    cout << "Penalty " << pen << endl;
    if (flag == 1)
        cout << "Withdraw penalty" << endl;
    else
        cout << "Balance after withdraw of " << with
        << "is " << bNW;
}
3;
class SovAcc: public Account
{
    int b, dep, with, bal, flag;
    double t, i;
    char c;
public:
    SovAcc(): Account()
    {
        cout << "Enter amount to be deposited" << endl;
        cin >> dep;
```

```
cout << "Enter the duration of account in years" << endl;
cin >> t;
cout << " Enter the amount for withdrawal" << endl;
cin >> with;
3
void totalBal();
void display();
3;
void savAcc :: totalBal()
{
    flag = 0;
    cout << "Do you want to deposit" << endl;
    cin >> c;
    if (c == 'Y')
        balance = balance + dep;
    else
        balance = balance + 0;
    i = (balance * (7.8) * t) / 100;
    balance = balance + i;
    if (bal < with)
        flag = 1;
    else
        bal = balance - with;
}
3
void savAcc :: display()
```

```
Topic :  
display A();  
cout << "Balance after deposit = " << balance << endl;  
if (flag == 1)  
    cout << "withdrawal not permitted" << endl;  
else  
    cout << "Balance after withdrawal of " << withdraw << " is "  
    << bal;  
}  
int main()  
{  
    int n;  
    cout << "1. Current Account in 2. Saving  
    Account << endl;  
    cin >> n;  
    if (n == 1)  
    {  
        CurrentAcc c;  
        c.add();  
        c.penalty();  
        c.display();  
    }  
    else  
    {  
        SavAcc s;  
        s.totalBal();  
        s.display();  
    }  
    return 0;  
}
```