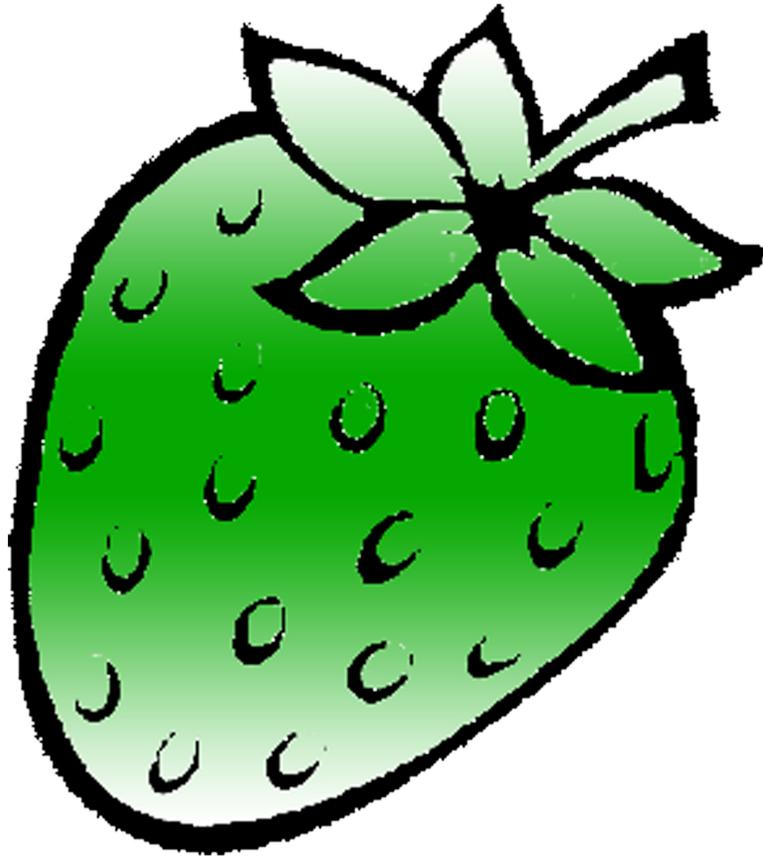


STRAWBERRY



/strawberrydevelopers



/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com

Experiment No. 3

PART A

(PART A: TO BE REFERRED BY STUDENTS)

A.1 Aim: To study below concepts of classes and objects

1. Array of Objects
2. Objects as a function argument
3. Static Members

P1: Define a class to represent a *bank account*. Include the following members:

Data Members:

Name of the depositor, Account number, type of account, balance

Member functions:

To assign initial values, to deposit an amount, To withdraw an amount and display new amount, To display name and balance.

Write a main program to test the program.

P2. Modify the class in P1 (Practical No 3), for handling 5 customers. The program must display count of users who had withdrawn money from the account.

A.2 Prerequisite:

Knowledge of classes, object and array.

A.3 Outcome:

After successful completion of this experiment students will be able to

1. Know use of classes and objects.
2. Know how to send object as input argument in functions.
3. Understand the use of static data member.

A.4 Theory:

Array of Objects:

Arrays of variables of type "class" are known as "**Array of objects**". The "identifier" used to refer the array of objects is an user defined data type.

Example:

```
#include <iostream>
```

```
class employee
```

Developed By Strawberry

```

{
    char name[30];
    float age;
    public:
        void getdata( )
        {
            cout << "Enter name:";
            cin >> name;
            cout << "Enter age:";
            cin >> age;
        }
        void putdata( )
        {
            cout << "Name:" << name << "\n";
            cout << "Age:" << age << "\n";
        }
};

const int size=3;

int main()
{
    employee manager[size];
    for(int i=0; i<size; i++)
    {
        cout << "\nDetails of manager:" << i+1 << "\n";
        manager[i].getdata();
    }
    for(i=0; i<size; i++)
    {
        cout << "\nManager: " << i+1 << "\n";
        manager[i].putdata( );
    }
    return 0;
}

```

Result:

Input:

Developed By Strawberry

Details of manager1

Enter name: xxx

Enter age: 45

Details of manager2

Enter name: yyy

Enter age: 37

Details of manager3

Enter name: zzz

Enter age: 50

Output:

Manager: 1

Name: xxx

Age: 45

Manager: 2

Name: yyy

Age: 37

Manager: 3

Name: zzz

Age: 50

In the above example an array of object "manager" is defined using the user defined data type "employee". The member function "getdata()" is used to get the input that is stored in this array of objects and "putdata()" is used to display the information.

Objects as function argument:

Similar to variables, objects can be passed on to functions. There are two methods to pass an object as an argument to a function as given below:

- (a) Pass-by-value: In this type a copy of the entire object (actual object) is sent to function and assigned to object of callee function (formal object). Both actual and formal copies of objects are stored at different memory locations. Hence, changes made in formal objects are not reflected to actual objects.
- (b) Pass-by-reference: In this type only the address of the object is sent to function. The formal argument is reference pointer to the actual object. Hence, changes made in the object are reflected to actual object. This method is useful because only address is passed to the function and duplicating of object is prevented.

Developed By Strawberry

Example:

```
#include <iostream>
class time
{
    int hours;
    int minutes;
public:
    void gettime(int h, int m)
    {
        hours=h;
        minutes=m;
    }
    void puttime(void)
    {
        cout << hours << " hours and " << minutes << " minutes" << "\n";
    }
    void sum(time, time); // declaration with objects as arguments
};

void time :: sum(time t1, timet2)
{
    minutes = t1.minutes + t2.minutes;
    hours = minutes/60;
    minutes = minutes%60;
    hours = hours + t1.hours + t2.hours;
}
int main()
{
    time T1, T2, T3;
    T1.gettime(2,45); //get T1
    T2.gettime(3,30); //get T2
    T3.sum(T1,T2); //T3=T1+t2
    cout << "T1 = "; T1.puttime(); // display T1
    cout << "T2 = "; T2.puttime(); // display T2
    cout << "T3 = "; T3.puttime(); // display T3
    return 0;
}
```

Developed By Strawberry

Output:

T1 = 2 hours and 45 minutes

T2 = 3 hours and 30 minutes

T3 = 6 hours and 15 minutes

Static Data Members:

A data member of a class can be qualified as static. The properties of a static member variable are similar to that of a C static variable. A static member variable has following characteristics.

- It is initialized to zero when the first object of its class is created. No other initialization is permitted.
- Only one copy of that member is created for the entire class and is shared by all the objects of that class, no matter how many objects are created.
- It is visible only within the class, but its lifetime is the entire program.

Static variables are normally used to maintain values common to the entire class. For example, a static data member can be used as a counter that records the occurrences of all the objects.

Program below illustrates the use of a static data member.

```
#include <iostream>
```

```
class item
{
    static int count;
    int number;
public:
    void getdata(int a)
    {
        number = a;
        count ++;
    }
    void getcount(void)
    {
        cout << "Count: ";
        cout << count << "\n";
    }
};
```

Developed By Strawberry

```
int item :: count;

int main()
{
    item a,b,c;
    a.getcount();
    b.getcount();
    c.getcount();
    a.getdata(100);
    b.getdata(200);
    c.getdata(300);
    cout << "After reading data"<<"\n";
    a.getcount();
    b.getcount();
    c.getcount();
    return 0;
}
```

Output:

```
Count: 0
Count: 0
Count: 0
After reading data
Count: 3
Count: 3
Count: 3
```

The type and scope of each static member variable must be defined outside the class definition. This is necessary because the static data members are stored separately rather than as a part of an object. Since they are associated with the class itself rather than with any class object, they are also known as class variables.

Developed By Strawberry

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)

Roll No. N008	Name: AKSHAY BANDA
Program: MBA TECh CS	Division: C
Semester: 2	Batch : C1
Date of Experiment: 21/1/2015	Date of Submission: 21/1/2015
Grade :	

B.1 Software Code written by student:

(Paste your C++ code completed during the 2 hours of practical in the lab here)

1.

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

```
class bank
{
private:
char n[25];
int an,a;
char t[25];
double b;
public:
void input()
{
cout<<"Enter name : ";
cin>>n;
cout<<"Enter Account number : ";
cin>>an;
cout<<"Enter account type : ";
cin>>t;
cout<<"Enter balance : ";
```

Developed By Strawberry

```

        cin>>b;
    }
    void deposit(double d)
    {
        b=b+d;
    }
    void withdraw(double w)
    {
        b=b-w;
    }
    void display()
    {
        cout<<"Name : "<<n;
        cout<<"Balance : "<<b;
    }
};

int main()
{
    int c;
    double d;
    bank b;
    b.input();
    do
    {
        cout<<"\nEnter 1 for deposit";
        cout<<"\nEnter 2 for withdraw";
        cout<<"\nEnter 3 for display";
        cout<<"\nAny other number to exit";
        cout<<"\nEnter your choice : ";
        cin>>c;
        if(c==1)
        {
            cout<<"Enter amount to deposit : ";
            cin>>d;
            b.deposit(d);
        }
        else if(c==2)
        {
            cout<<"Enter amount to withdraw : ";

```

Developed By Strawberry

```

        cin>>d;
        b.withdraw(d);
    }
    else if(c==3)
    {
        b.display();
    }
}while(c>=1 && c<=3);
return 0;
}

```

2.

```

#include<iostream>
using namespace std;

```

```

class bank
{
    private:
        static int c;
        char n[25];
        int an;
        char t[25];
        double b;
    public:
        void input()
        {
            cout<<"Enter name : ";
            cin>>n;
            cout<<"Enter Account number : ";
            cin>>an;
            cout<<"Enter account type : ";
            cin>>t;
            cout<<"Enter balance : ";
            cin>>b;
        }
        int check(int s, int i)
        {
            if(s==an)
            {
                return i;
            }
        }
    }

```

Developed By Strawberry

```

    }
    else
        return -1;
}
void deposit(double d)
{
    b=b+d;
}
void withdraw(double w)
{
    b=b-w;
    c++;
}
void display()
{
    cout<<"\nName : "<<n;
    cout<<"\nBalance : "<<b;
}
void count()
{
    cout<<"Number of withdraw is "<<c;
}
};
int bank::c;

int main()
{
    int n=1,c,i,a,t=-1;
    double d;
    cout<<"\nEnter number of customers : ";
    cin>>n;
    bank b[n];
    for(i=0;i<n;i++)
    {
        b[i].input();
    }
    do
    {
        cout<<"Enter account number or press 0 to exit : ";
        cin>>a;

```

Developed By Strawberry

```

for(i=0;i<n;i++)
{
    t=b[i].check(a,i);

    if(t!=-1)
        break;
}
if(t>=0 && t<=n)
{
    c=1;
    while(c>=1 && c<=3)
    {
        cout<<"\nEnter 1 for deposit";
        cout<<"\nEnter 2 for withdraw";
        cout<<"\nEnter 3 for display";
        cout<<"\nAny other number to exit";
        cout<<"\nEnter your choice : ";
        cin>>c;
        if(c==1)
        {
            cout<<"Enter amount to deposit : ";
            cin>>d;
            b[t].deposit(d);
        }
        else if(c==2)
        {
            cout<<"Enter amount to withdraw : ";
            cin>>d;
            b[t].withdraw(d);
        }
        else if(c==3)
        {
            b[t].display();
        }
    }
    t=-1;
}
else
{
    cout<<"Invalid account number";
}

```

Developed By Strawberry

```
    }  
    }while(a!=0);  
    b[0].count();  
    return 0;  
}
```

B.2 Input and Output:

(Paste your program input and output in following format. If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)

1.

Enter name : akshay

Enter Account number : 718

Enter account type : sa

Enter balance : 10000

Enter 1 for deposit

Enter 2 for withdraw

Enter 3 for display

Any other number to exit

Enter your choice : 1

Enter amount to deposit : 10000

Enter 1 for deposit

Enter 2 for withdraw

Enter 3 for display

Any other number to exit

Enter your choice : 2

Enter amount to withdraw : 2000

Enter 1 for deposit

Enter 2 for withdraw

Enter 3 for display

Any other number to exit

Enter your choice : 3

Name : akshayBalance : 18000

Enter 1 for deposit

Developed By Strawberry

Enter 2 for withdraw
Enter 3 for display
Any other number to exit
Enter your choice : 4

2.

Enter number of customers : 2
Enter name : akshay
Enter Account number : 1
Enter account type : sas
Enter balance : 10000
Enter name : kd
Enter Account number : 2
Enter account type : sas
Enter balance : 2000
Enter account number or press 0 to exit : 1

Enter 1 for deposit
Enter 2 for withdraw
Enter 3 for display
Any other number to exit
Enter your choice : 1
Enter amount to deposit : 2000

Enter 1 for deposit
Enter 2 for withdraw
Enter 3 for display
Any other number to exit
Enter your choice : 3

Name : akshay
Balance : 12000
Enter 1 for deposit
Enter 2 for withdraw
Enter 3 for display
Any other number to exit
Enter your choice : 0
Enter account number or press 0 to exit : 2

Developed By Strawberry

Enter 1 for deposit
Enter 2 for withdraw
Enter 3 for display
Any other number to exit
Enter your choice : 3

Name : kd
Balance : 2000
Enter 1 for deposit
Enter 2 for withdraw
Enter 3 for display
Any other number to exit
Enter your choice : 0
Enter account number or press 0 to exit : 0
Invalid account numberNumber of withdraw is 0
Process returned 0 (0x0) execution time : 39.125 s
Press any key to continue.

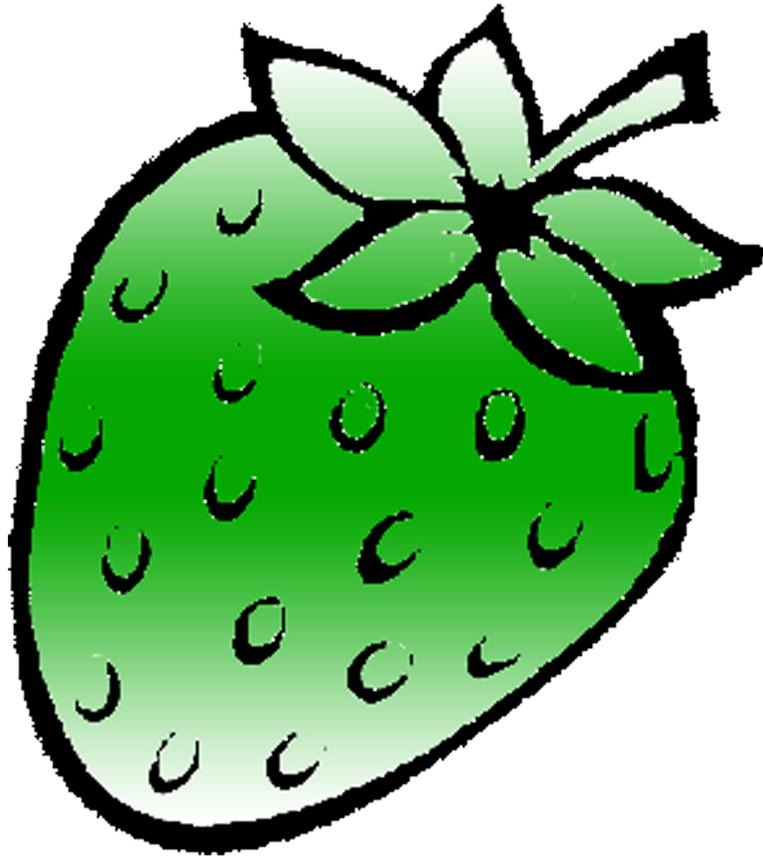
B.3 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.1)

I learned the use of classes and objects, to send object as input argument in functions, use of static data member.

Developed By Strawberry

STRAWBERRY



/strawberrydevelopers



/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com