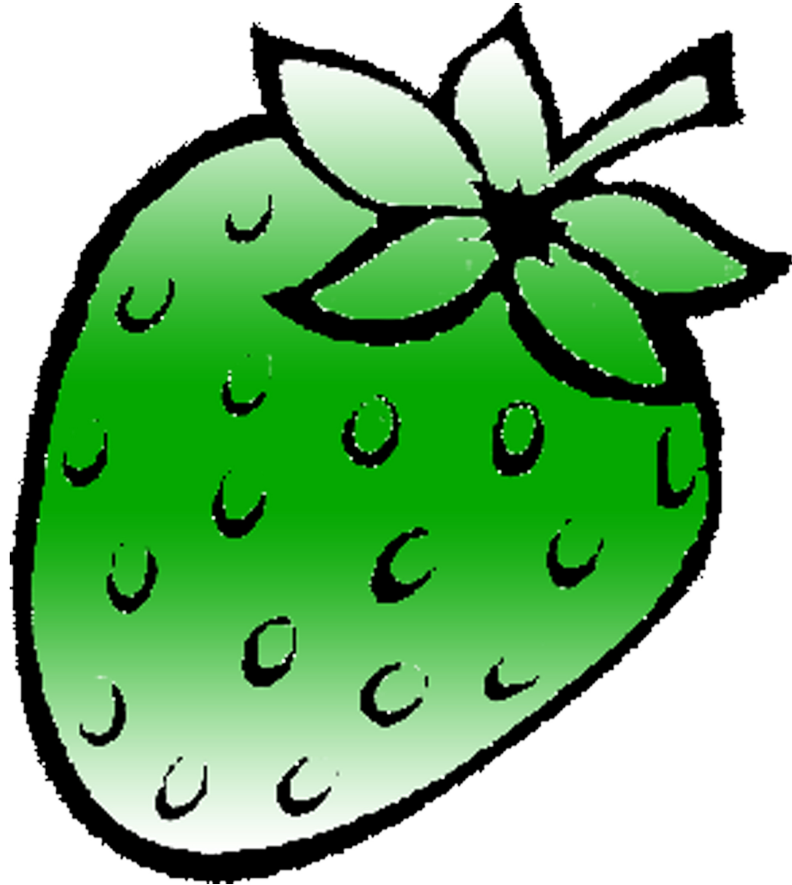


STRAWBERRY



/strawberrydevelopers



/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com

Experiment No. 4

PART A

(PART A: TO BE REFERRED BY STUDENTS)

A.1. Aim: To understand the following concept of constructors.

1. To create constructor.
2. To understand concept of no argument constructor.
3. To understand the concept of default constructor.
4. To understand the concept of parameterized constructor.

P1: Create a class called *time* that has separate integer member data for hours, minutes, and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. Another member function should display it, in HH:MM:SS format. The final member function should add two objects of type *time* passed as arguments.

A `main()` program should create two initialized *time* objects and one that isn't initialized. Then it should add the two initialized values together, leaving the result in the third *time* variable. Finally it should display the value of this third variable. Make appropriate member functions.

A.2. Prerequisite:

Sr.No	Concepts
1.	Constructors

A.3. Outcome:

After successful completion of this experiment students will be able to

A.4. Theory:

Constructors:

A constructor is a 'special' member function whose task is to initialize the objects of its class. It is special because its name is the same as the class name. The constructor is invoked whenever an object of its associated class is created. It is called constructor because it constructs the values of the data members of the class.

A constructor is declared and defined as follows:

```
//class with a constructor
class xyz
{
    int m, n;
```

Developed By Strawberry

```

    public:
        xyz(void);          // constructor declared
        .....
};
xyz :: xyz(void)          // constructor defined
{
    m=0;
    n=0;
}

```

When a class contains a constructor it is guaranteed that an object created by the class will be initialized automatically. For example, the declaration

```
xyz obj1;          // object obj1 created
```

not only creates the object obj1 of type xyz but also initializes its data members m and n to zero.

A constructor that accepts no parameters is called the **default constructor**. The default constructor for class A is A::A(). If no such constructor is defined, then the compiler supplies a default constructor. Therefore, a statement such as

```
A a;
```

Invokes the default constructor of the compiler to create the object a.

The constructor functions have the following characteristics:

- They should be declared in public section
- They are invoked automatically when the objects are created.
- They do not have return types not even void, and therefore, they cannot return values.
- They cannot be inherited, though a derived class can call the base class constructor.
- Like other C++ functions, they can have default arguments.
- Constructors cannot be virtual
- We cannot refer to their addresses
- An object with a constructor cannot be used as a member of a union.
- They make ‘implicit calls’ to the operators new and delete when memory allocation is required.

Parameterized constructor:

The constructors that can take arguments are called parameterized constructors.

The constructor xyz() may be modified to take arguments as shown below:

```
class xyz
{
    int m, n;
    public:

```

Developed By Strawberry

```

        xyz(int x, int y);    // parameterized constructor
        .....
};
xyz :: xyz(int x, int y)
{
    m=x;
    n=y;
}

```

When a constructor has been parameterized, we must pass the initial values as arguments to the constructor function when an object is declared. This can be done in two ways:

- By calling the constructor explicitly.
`xyz obj1 = xyz(0,100);` // explicit call
- By calling the constructor implicitly.
`xyz obj1(0,100);` // implicit call

Multiple constructors in a class:

C++ permits the use of multiple constructors in the same class. For example, we could define a class as follows:

```

class xyz
{
    int m, n;
    public:
        xyz()                // constructor 1
        {
            m=0;
            n=0;
        }
        xyz(int x, int y)    // constructor 2
        {
            m=x;
            n=y;
        }
        xyz( xyz &i)        // constructor 3
        {
            m=i.m;
            n=i.n;
        }
};

```

Developed By Strawberry

This declares three constructors for a xyz object. The first constructor receives no arguments, the second receives two integer arguments and the third receives one xyz object as an argument. For example, the declaration

```
xyz obj1;
```

would automatically invoke the first constructor and set both m and n of obj1 to zero. The statement

```
xyz obj2(20,40);
```

would call the second constructor and set both m and n of obj2 to 20 and 40 respectively. Finally the statement

```
xyz obj3(obj2);
```

would invoke the third constructor which copies the values of obj2 into obj3. In other words, it sets the value of every data element of obj3 to the values of the corresponding data element of obj2. Such a constructor is called the **copy constructor**.

When more than one constructor function is defined in a class, we say that the constructor is overloaded.

Constructor with default arguments:

It is possible to define constructors with default arguments.

For example, the constructor xyz() can be declared as follows:

```
xyz( int x, int y=0);
```

the default value of the argument y is zero. Then, the statement

```
xyz obj(5);
```

Assigns the value 5 to m and 0 to n (by default).

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 the 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: 28/1/15	Date of Submission: 28/1/15
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 time
{
private:
    int h,m,s;
public:
    time()
    {
        h=0;
        m=0;
        s=0;
    }
    time(int a, int b, int c)
    {
        h=a;
        m=b;
        s=c;
    }
    time add(time a, time b)
    {
        time c;
        c.h=a.h+b.h;
        c.m=a.m+b.m;
        c.s=a.s+b.s;
        if(c.s>=60)
```

Developed By Strawberry

```

    {
        c.m=m+(c.s/60);
        c.s=c.s%60;
    }
    if(c.m>=60)
    {
        c.h=c.h+(c.m/60);
        c.m=c.m%60;
    }
    return c;
}
void display()
{
    cout<<"Time is "<<h<<":"<<m<<":"<<s;
}
};

```

```

int main()
{
    int h,m,s;
    cout<<"\nEnter hours :";
    cin>>h;
    cout<<"\nEnter minutes : ";
    cin>>m;
    cout<<"\nEnter seconds : ";
    cin>>s;
    time x(h,m,s);

    cout<<"\nEnter hours :";
    cin>>h;
    cout<<"\nEnter minutes : ";
    cin>>m;
    cout<<"\nEnter seconds : ";
    cin>>s;
    time y(h,m,s);

    time z;
    z=z.add(x,y);
    z.display();
}

```

B.2. Input and Output:

Developed By Strawberry

(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 hours :7

Enter minutes : 7

Enter seconds : 7

Enter hours :18

Enter minutes : 18

Enter seconds : 18

Time is 25:25:25

Process returned 0 (0x0) execution time : 4.188 s

Press any key to continue.

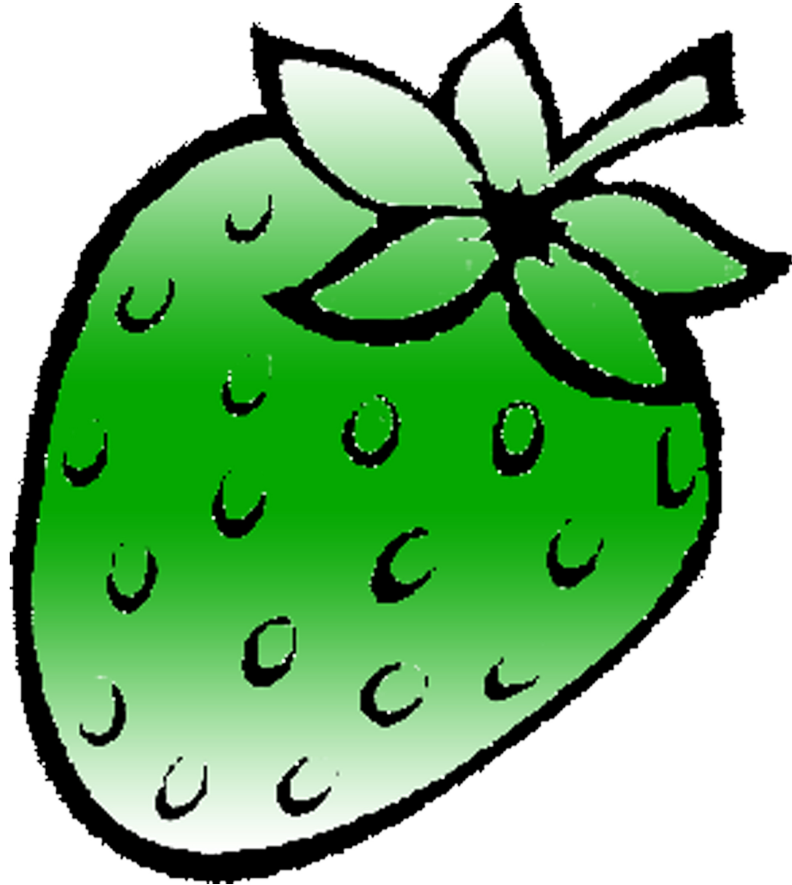
B.3. Conclusion:

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

I learned to create constructor, understand concept of no argument constructor, concept of default constructor, concept of parameterized constructor.

Developed By Strawberry

STRAWBERRY



/strawberrydevelopers



/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com