Writing Classes

Contents

- Difference between class and structure
- Define class
- How create class
- How to write constructor and its rules.

Difference between class and structure

- We can use structure in C++.
- By default structure data members and functions are public.
- But as per C++ terminology we need to use here class instead of struct keyword.
- By default all members of class are private.
- class and object are C++ terminology.

class

- Template for creation of similar type of objects.
- A class in C++ supports Data abstraction and encapsulation.

```
class classname
{
        private:
                variable declaration;
        public:
                function declartion;
};
```

Class cDate

```
class cDate
{
        private:
                int Day,Month,Year;
        public:
                void accept();
                void display();
};
```

Class components

- **Access Specifiers**
    1. private
    2. public
    3. protected
- **Data Members**
- **Member Functions**
    1. Constructors
    2. Destructors
    3. Ordinary member functions

Types of Constructors

- **Default Constructor**:

A constructor having no arguments called no argument constructor and default constructor.

- **Parameterized Constructor**:

A constructor having arguments called parameterized constructor.

Rules of for creating constructor

1. Used same name as class name.
2. No return type even void.

Example

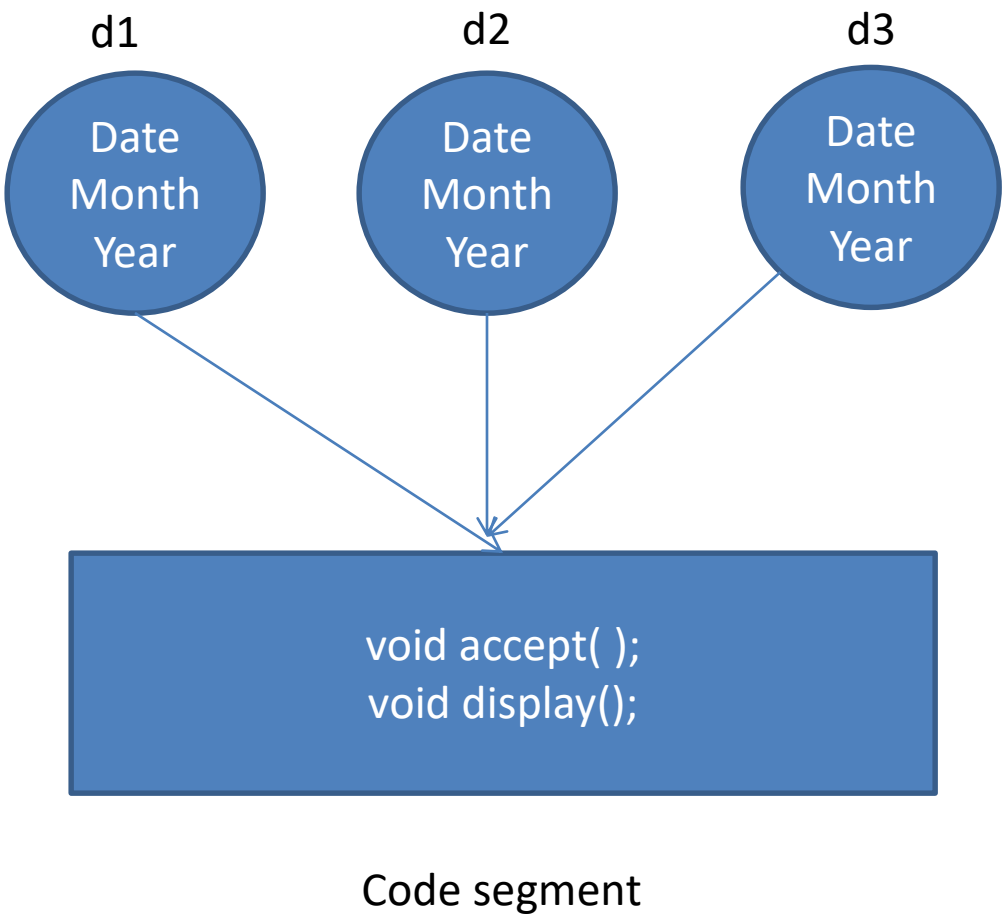
```
cDate( )   //Default
{
        Date=1;
        Month=1;
        Year=2018;
}

cDate(int d , int m , int y)  //Parameterized
{
        Date=d;
        Month=m;
        Year=y;
}
```

```
int main()
{
        cDate d1;
        cDate d2(1,1,2020);
}
```

Types of member functions

- **Mutator functions-** Changes the contents of data members.
- **Accessor functions-**Accesses the instance members.
- **Facilitator functions-**Helps to view the data of object.

Memory allocation to objects



This Keyword

- This is keyword in C++.
- It is hidden parameter.
- It will store the address of current object.
- Need to explicitly mention when data member and local variables names are same.

How to create object on heap

```
int main()
{
        cDate *ptr;
        ptr=new cDate(1,1,2018);

        delete ptr;
}
```

Constant member function

```
class cDate
{
        ……….
        public:
                void display() const;
}

Void cDate :: display ( ) const
{
        ……….
}
```

Static variables

- Need to use when need to provide common value among the all object.
- E.g. interest rate in bank,counting no of objects.
- Allocated memory once and every object can share it.

Default Arguments to function

- Function can be declared with default values.
- C++ allows calling function without specifying all its arguments if the function is declared with default values.
- Default values must be specified in declaration and not in definition.
- The values must mentioned from right hand side.

Default Arguments to function: Example

```
void display(int , int = 20, int = 30);

void display( int x, int y, int z)
{
        cout<<x;
        cout<<y;
        cout<<z;
}
int main()
{
        display();    //error
        display(10);  //display 10,20,30
        display(10,50);  //display 10,50,30
        display(10, , 30); //error
}
```

Lab Assignments

- Create a cDate class with follwing
    1. data members-date,month,year.
    2. Constructors both.
    3. accept and display function

- Create cComplex class which having following
    1. data members: real and imag no.
    2. constructors
    3. accept and display function.