Lab-5

Topic: Constructors and Destructors

Create a class complex which stores real and imaginary part of a complex number. Include all types of constructors and destructor. The destructor should display a message about the destructor being invoked. Create objects using different constructors and display them.

```
Class complex{
  Int real;
   Int img;

Public:
   //Complex(){}
Complex()
{  real=;
   Img=;
}
Complex (int I, int j)
{  }
Complex (const complex & c1)
{  real=c1.real;
   Img=c1.img;}
Void display(){
  }
};
Int main(){
Complex c1, c2(4,5);
  Complex c3(c2);
C1.display();



}
```

Create a class which stores time in hh:mm format. Include all the constructors. The parameterized constructor should initialize the minute value to zero, if it is not provided.

Create a class which stores a string and its length as data members. Include all the constructors. Include a member function to join two strings and display the concatenated string.

```
  Class mystr{
  Char *p;
Int len;
Public:
  Mystr()
{
  P=new char;
     Len=0;
     }
     Mystr(char *s, int n)
     {
      Len=n;
       P=new char[len+1];
       Strcpy(p,s);
     }
```

iv. Write a C++ program using class to dynamically allocate two integer arrays,

add them it to a third array and display all the arrays.

```
For(int I=0; I<a.len;I++)

  P[I]=a.p[I];

1 3 5   7 6 4

1 3 5 7 6 4
```

V. WAP to demonstrate the order of call of constructors and destructors for a class.

Vi .WAP to count number of objects created from a class using concept of static data members and static member function.

```
Class test{
  Int I;
  Static int count;
Public:
  Test(){count++;}
  Test(int k) { I=k; count ++;}
 Static void print(){ cout<<”   “<<count; }
};
  int test:: count;
Int main()
{
```

Vii. A book shop maintains the inventory of books that are being sold at the workshop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies are available, the total cost of their quested copies is displayed otherwise the message “Required copies not in stock” is displayed. WAP using a class called Books with suitable member functions and constructors.

```
Void search( book *b, int len, char * a, char *t)

{ int no;

  For(I

   If ((Strcmp(b[I].author, a)==0 &&strcmp(b[I].title, t)==0)

      {
             Cout<<”Enter the no of copies “;
            Cin>>no;
             If(b[I].stock>=no)
              {  int bill=no * b[I].price;
                 Cout<<;
                 B[I].stock=-no;
              }
              Else
               Cout<<

            Else
              Cout<<”not found”
```