Chapter-8
Constructors and Destructors

Examples:

1.[Pg-185]

```
# include<iostream.h>
# include<conio.h>
class simple
{
private:
int a,b;
public:
simple()
{
a=0;
b=0;
cout<<"\n Constructor of class-
simple…";
}
~simple()
{
cout<<"\n Destructor ofclass-
simple…";
}
void getdata()
{
cout<<"\n Enter valuesfor a and b";
cin>>a>>b;
}
void putdata()
{
cout<<"\n The two
integers…"<<a<<'\t'
                              <<b;
cout<<"\n The sum of the variables…"
    <<a+b;
}
};
void main()
{
simple s;
s.getdata();
s.putdata();
}
```

Output:

```
Constructor of class-simple...
Enter values for a and b...5 6
The two integers...5 6
The sum of the variables...11
Destructor of class-simple...
```

2.[Pg-187]

```
# include<iostream.h>
# include<conio.h>
class add
{
int num1,num2,sum;
public:
add()
{
cout<<"\n Constructor without
          parameters…";
num1=0;
num2=0;
sum=0;
}
add(int s1,int s2)
{
cout<<"\n Parameterized
constructor";
num1=s1;
num2=s2;
sum=NULL;
}
add(add &a)
{
cout<<"\n Copy constructor…";
num1=a.num1;
num2=a.num2;
sum=NULL;
}
void getdata()
{
cout<<"Enter data…";
cin>>num1>>num2;
}
void addition()
{
sum=num1+num2;
}
void putdata()
{
cout<<"\n The numbers are…";
cout<<num1<<'\t'<<num2;
cout<<"\n The sum of the numbers
      are…"<<sum;
}
};
void main()
{
clrscr();
add a,b(10,20),c(b);
```

```
a.getdata();
a.addition();
b.addition();
c.addition();
cout<<"\n Object a:";
a.putdata();
cout<<"\n Object b:";
b.putdata();
cout<<"\n Object c:";
c.putdata();
}
```

Output:

```
Constructor without parameters..
Parameterized constructors...
Copy constructors...
Enter data...5 6
Object a:
The numbers are...5 6
The sum of the numbers are...11
Object b:
The numbers are...10 20
The sum of the numbers are...30
Object c:
The numbers are...10 20
The sum of the numbers are...30
```

3.[Pg-190]

```
# include<iostream.h>
# include<conio.h>
class add
{
int num1,num2,sum;
public:
add()
{
cout<<"\n Constructor without
          parameters…";
num1='\0';
num2='\0';
sum='\0';
}
add(int s1,int s2)
{
cout<<"\n Parameterized
constructor";
num1=s1;
num2=s2;
sum=NULL;
}
add(add &a)
{
cout<<"\n Copy constructor…";
num1=a.num1;
num2=a.num2;
sum=NULL;
}
void getdata()
{
cout<<"Enter data…";
cin>>num1>>num2;
}
void addition(add b)
{
sum=num1+num2+b.num1+b.num2;
}
add addition()
{
add a(5,6);
sum=num1+num2+a.num1+a.num2;
}
void putdata()
{
cout<<"\n The numbers are…";
cout<<num1<<'\t'<<num2;
cout<<"\n The sum of the numbers
    are…"<<sum;
}
};
void main()
{
clrscr();
add a,b(10,20),c(b);
a.getdata();
a.addition(b);
b=c.addition();
c.addition();
cout<<"\n Object a:";
a.putdata();
cout<<"\n Object b:";
b.putdata();
cout<<"\n Object c:";
c.putdata();
}
```

Output:

```
Constructor without parameters...
Parameterized constructor...
Copy constructor...
Enter data..2 3
Copy constructor...
Parameterized constructor...
Parameterized constructor...
Object a:
The numbers are...2 3
The sum of the two numbers...35
Object b:
The numbers are...1 1494
The sum of the two numbers...0
Object c:
The numbers are...10 20
The sum of the two numbers...41
```

Exercise:

1.Complete the table:[Pg-194]

	Constructor	Destructor
1.Should be declared under	Public scope	Public scope
2.Overloading is	allowed	not allowed
3.Is executed when an object is	created	Destroyed
4.The function is	To initialize the object and allocates memory.	Remove the memory and values in the memory.

2.Why do the following snippets show errors? [Pg-194]

a)
```
class simple
  {
   private:
   int x;
   simple()
   {x=5;}
  };
```

Error:

Constructor cannot be declared under private. I should be declared under public.

b)
```
class simple
  {
   private:
   int x;
   public:
   simple(int y)
   {x=y;}
  };
  void main()
  {
   simple s;
  }
```

Error:

Cannot create an object s as it does not match the constructor.

c)
```
class simple
{
 private:
 int x;
public:
 simple(int y)
 {x=y;}
simple(int z=5)
{
 x=z;
}
};
void main()
{
 simple s(6);}
```

Error:

The two prototypes are similar as they do not differ either in value or by type.