

**DRONACHARYA GROUP OF INSTITUTIONS, GREATER NOIDA**

**Affiliated to Uttar Pradesh Technical University, Lucknow  
Approved by AICTE**



DEPARTMENT OF INFORMATION TECHNOLOGY

**Lab Manual for  
Advanced Programming Lab (NCS-355)**

## **List of Experiments**

1. Programs using Functions and Pointers in C
2. Programs using Files in C
3. Programs using Classes and Objects
4. Programs using Operator Overloading
5. Programs using Inheritance, Polymorphism and its types
6. Programs using Arrays and Pointers
7. Programs using Dynamic memory allocation
8. Programs using Templates and Exceptions
9. Programs using Sequential and Random access files

## LAB OBJECTIVE

- Overview of C and C++ Language
- Basic Concept of Object Oriented Programming
- Advance Programming
- C++ is a statically typed, compiled, general-purpose, case-sensitive, free-form programming language that supports procedural, object-oriented, and generic programming.
- C++ is regarded as a **middle-level** language, as it comprises a combination of both high-level and low-level language features.
- C++ was developed by Bjarne Stroustrup starting in 1979 at Bell Labs in Murray Hill, New Jersey, as an enhancement to the C language and originally named C with Classes but later it was renamed C++ in 1983.
- C++ is a superset of C, and that virtually any legal C program is a legal C++ program.

## Guidelines to Students

---

- Equipment in the lab for the use of student community. Students need to maintain a proper decorum in the computer lab. Students must use the equipment with care. Any damage is caused is punishable.
- Students are required to carry their observation / programs book with completed exercises while entering the lab.
- Students are supposed to occupy the machines allotted to them and are not supposed to talk or make noise in the lab. The allocation is put up on the lab notice board.

- Lab can be used in free time / lunch hours by the students who need to use the systems should take prior permission from the lab in-charge.
- Lab records need to be submitted on or before date of submission.
- Students are not supposed to use floppy disks

## Experiment 1

### Write a program in C using Function and Pointers

```
#include <stdio.h>
void swap(int *a,int *b);
int main(){
    int num1=5,num2=10;
    swap(&num1,&num2); /* address of num1 and num2 is passed to swap function */
    printf("Number1 = %d\n",num1);
    printf("Number2 = %d",num2);
    return 0;
}
void swap(int *a,int *b){ /* pointer a and b points to address of num1 and num2
respectively */
    int temp;
    temp=*a;
    *a=*b;
    *b=temp;
}
```

### OUTPUT :

```
Number1= 10
Number2= 5
```

## Experiment 2

### Write a program to merge the contents of two files and write into third file

```
#include <stdio.h>
#include <stdlib.h>

main()
{
    FILE *fsring1, *fsring2, *ftemp;
    char ch, file1[20], file2[20], file3[20];

    printf("Enter name of first file ");
    gets(file1);
    printf("Enter name of second file ");
    gets(file2);
    printf("Enter name to store merged file ");
    gets(file3);
    fsring1 = fopen(file1, "r");
    fsring2 = fopen(file2, "r");
    if (fsring1 == NULL || fsring2 == NULL)
    {
        perror("Error has occurred");
        printf("Press any key to exit...\n");
        exit(EXIT_FAILURE);
    }
    ftemp = fopen(file3, "w");
    if (ftemp == NULL)
    {
        perror("Error has occurs");
        printf("Press any key to exit...\n");
        exit(EXIT_FAILURE);
    }
    while ((ch = fgetc(fsring1)) != EOF)
        fputc(ch, ftemp);
    while ((ch = fgetc(fsring2)) != EOF)
        fputc(ch, ftemp);
    printf("Two files merged %s successfully.\n", file3);
```

```
fclose(fsring1);
fclose(fsring2);
fclose(ftemp);
return 0;
}
```

### Experiment 3

#### Programs using Classes and Objects

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

Data Members

- 1 Name of the depositor
- 2 Account Number
- 3 Type of account
- 4 Balance amount in the account

Member function

- 1 To assign initial values
- 2 To deposit an amount
- 3 To withdraw an amount after checking the balance
- 4 To display name and Balance

write a main program to test the program.

```
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
```

```
class bank
{
char name[20];
int acno;
char actype[20];
int bal;
public :
void opbal(void);
void deposit(void);
```

```
void withdraw(void);  
void display(void);  
};
```

```
void bank :: opbal(void)  
{  
cout<<endl<<endl;  
cout<<"Enter Name :-";  
cin>>name;  
cout<<"Enter A/c no. :-";  
cin>>acno;  
cout<<"Enter A/c Type :-";  
cin>>actype;  
cout<<"Enter Opening Balance:-";  
cin>>bal;  
}
```

```
void bank :: deposit(void)  
{  
cout<<"Enter Deposit amount :-";  
int deposit=0;  
cin>>deposit;  
deposit=deposit+bal;  
cout<<"\nDeposit Balance = "<<deposit;  
bal=deposit;  
}
```

```
void bank :: withdraw(void)  
{  
int withdraw;  
cout<<"\nBalance Amount = "<<bal;  
cout<<"\nEnter Withdraw Amount :-";  
cin>>withdraw;  
bal=bal-withdraw;  
cout<<"After Withdraw Balance is "<<bal;  
}
```

```
void bank :: display(void)  
{  
cout<<endl<<endl<<endl;
```



```
cout<<setw(50)<<"DETAILS"<<endl;
cout<<setw(50)<<"name    "<<name<<endl;
cout<<setw(50)<<"A/c. No.  "<<acno<<endl;
cout<<setw(50)<<"A/c Type
```

```
"<<actype<<endl;
cout<<setw(50)<<"Balance  "<<bal<<endl;
}
```

```
void main()
{
clrscr();
bank o1;
int choice;
do
{
cout<<"\n\nChoice List\n\n";
cout<<"1) To assign Initial
Value\n";
cout<<"2) To Deposit\n";
cout<<"3) To Withdraw\n";
cout<<"4) To Display All Details
```

```
\n";
cout<<"5) EXIT\n";
cout<<"Enter your choice :-";
cin>>choice;
switch(choice)
{
case 1: o1.opbal();
break;
case 2: o1.deposit();
break;
case 3: o1.withdraw();
break;
case 4: o1.display();
break;
case 5: goto end;
}
}
```

```
}while(1);  
end:  
}
```

## **OUTPUT :**

Choice List

- 1) To assign Initial Value
- 2) To Deposit
- 3) To Withdraw
- 4) To Display All Details
- 5) EXIT

Enter your choice:-1

Enter name :Diya

Enter A/c No:101010101

Enter A/c type:Current

Enter Opening Balance:20000

Choice List

- 1) To assign Initial Value
- 2) To Deposit
- 3) To Withdraw
- 4) To Display All Details
- 5) EXIT

Enter your choice:-4

Name: Diya

A/c No: 101010101

A/c type: Current

Balnce : 20000

## Experiment 4

### Programs using Operator Overloading

```
#include<iostream.h>
#include<conio.h>

class complex
{
    int a,b;
public:
    void getvalue()
    {
        cout<<"Enter the value of Complex Numbers a,b:";
        cin>>a>>b;
    }
    complex operator+(complex ob)
    {
        complex t;
        t.a=a+ob.a;
        t.b=b+ob.b;
        return(t);
    }
    complex operator-(complex ob)
    {
        complex t;
        t.a=a-ob.a;
        t.b=b-ob.b;
        return(t);
    }
    void display()
    {
        cout<<a<<"+"<<b<<"i"<<"\n";
    }
};

void main()
{
    clrscr();
```

```

complex obj1,obj2,result,result1;

obj1.getvalue();
obj2.getvalue();

result = obj1+obj2;
result1=obj1-obj2;

cout<<"Input Values:\n";
obj1.display();
obj2.display();

cout<<"Result:";
result.display();
result1.display();

getch();
}

```

### **OUTPUT :**

```

Enter the value of Complex Numbers a, b
4          5
Enter the value of Complex Numbers a, b
2          2
Input Values
4 + 5i
2 + 2i
Result
6 + 7i
2 + 3i

```

## Experiment 5

### Programs using Inheritance, Polymorphism and its types

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class account{
intacno;
char name[20],type[10];
public:
voidacdetails(void);
void status(void);
};
void account::acdetails(void)
{
cout<<"\nEnter name:- ";
cin>>name;
cout<<"\nAccount number:- ";
cin>>acno;
cout<<"\nAccount type:- ";
cin>>type;
}
void account::status(void)
{
cout<<"\n\nName:- "<<name;
cout<<"\n\nAccount number:- "<<acno;
cout<<"\n\nAccount type:- "<<type;
}

class savacc:public account{
intdv,wv,in,fv,time;
public:
savacc(){
dv=0;
wv=0;
fv=0;
}
voiddiposit(void);
```

```

void withdraw(void);
void display(void);
};
void savacc::diposit(void)
{
acdetails();
cout<<"\nEnter deposit ammount in Rs.:- ";
cin>>dv;
cout<<"\nEnter time in years:- ";
cin>>time;
fv=fv+dv;
cout<<"\nYour balance is:- "<<fv;
in=time*8*fv/100;
cout<<"\n\nYou will get interest of Rs.:- "<<in;
fv=fv+in;
cout<<"\n\nAfter adding your interest your balance is Rs.:-"<<fv;
}
void savacc::withdraw(void)
{
acdetails();
cout<<"\nEnterwithdrawlammount in Rs.:- ";
cin>>wv;
fv=fv-wv;
cout<<"\n\nAfterwithdrawl your balance is Rs:-"<<fv;
}
void savacc::display(void)
{
status();
cout<<"\n\nAvailable balance is Rs:- "<<fv;
}
class curacc:public account{
int dv,wv,fv,cq,n;
public:
curacc(){
dv=0;
wv=0;
fv=0;
cq=0;
}
void diposit(void);

```

```

void withdraw(void);
void display(void);
};
void curacc::diposit(void)
{
acdetails();
cout<<"\n\nEnter deposit ammount in Rs.:- ";
cin>>dv;
fv=fv+dv;
cout<<"\n\nYour balance is:- "<<fv;
}
void curacc::withdraw(void)
{
acdetails();
cout<<"\n\nPress 1 to withdraw using cheque... ";
cout<<"\n\nPress 2 to withdraw as usual... ";
cout<<"\n\nEnter your choice:- ";
cin>>n;
if(n==1)
{
cout<<"\n\nEnter the cheque number:- ";
cin>>cq;
cout<<"\n\nEnter withdrawlammount in Rs.:- ";
cin>>wv;
fv=fv-wv;
}
else
{
cout<<"\n\nEnter withdrawlammount in Rs.:- ";
cin>>wv;
fv=fv-wv;
}
if(fv<1000)
{
cout<<"\n\nYou have no sufficient balance. Your fine is Rs. :- 20";
fv=fv-20;
}
cout<<"\n\nAfter withdrawl your balance is Rs:- "<<fv;
}
void curacc::display(void)

```

```

{
status();
cout<<"\n\nAvailable balance is Rs:- "<<fv;
}
void main(){
savacc s;
curacc c;
intx,y;
//clrscr();
do{
cout<<"\nEnter your choice... ";
cout<<"\n-----";
cout<<"\nPress 1 for savings account...";
cout<<"\nPress 2 for current account...";
cout<<"\nAny digit to exit...\n";
cout<<"\n-----\n\n";
cin>>x;
if (x==1)
{
do{
cout<<"\n\nYou are now in Savings account...";
cout<<"\n-----";
cout<<"\nPress 1 to deposit...";
cout<<"\nPress 2 to withdraw...";
cout<<"\nPress 3 to see status...";
cout<<"\nAny digit to back in previous menu...";
cout<<"\n-----\n\n";
cin>>y;
switch(y){
case 1:{
s.diposit();
break;
}

case 2:{
s.withdraw();
break;
}
case 3:{
s.display();

```



```

break;
}
default: {
break;
}
}
}
while(y<4);
}
else if(x==2)
{
do{
cout<<"\n\nYou are now in Current account...";
cout<<"\n-----";
cout<<"\nPress 1 to deposit...";
cout<<"\nPress 2 to withdraw...";
cout<<"\nPress 3 to see status...";
cout<<"\nAny digit to back in previous menu...";
cout<<"\n-----\n\n";
cin>>y;
switch(y){
case 1:{
c.diposit();
break;
}
case 2:{
c.withdraw();
break;
}
case 3:{
c.display();
break;
}
default: {
break;
}
}
}
while(y<4);
}

```

```
else
{
exit(0);
}
}
while(x<3);
getch();
}
```

Output

Enter your choice---  
Press 1 for savings account---  
Aditi Paul  
Press 2 for current account---  
Any digit to exit---2

-----  
You are in Current account---  
Press 1 to deposit---  
Press 2 to withdraw---  
Press 3 to see status---  
Any digit to back in previous menu---  
1  
Enter name:-Debajyoti  
Account number:-101010  
Account type:-Current  
Enter deposit amount in Rs:-1000  
Your balance is:-1000  
Press 1 to deposit---  
Press 2 to withdraw---  
Press 3 to see status---  
Any digit to back in previous menu---

## Experiment 6

### Programs using Arrays

#### Program to read a matrix and display it using array.

```
#include<stdio.h>
#include<conio.h>
```

```

#include<iostream.h>
class student{
inti,j,m,n,a[10][10];
public:
voidgetdata(void);
void display(void);
};
void student::getdata(void)
{
cout<<"Enter the order :-";
cin>>m>>n;
cout<<"Enter the elements:-\n";
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
cin>>a[i][j];
}
}
}
void student::display(void)
{
cout<<"\nYour matrix is---\n";
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
cout<<a[i][j];
cout<<"\t";
}
cout<<"\n";
}
}
int main()
{
//clrscr();
student s1;
s1.getdata();
s1.display();
getch();
}

```

```
return(0);  
}
```

## **OUTPUT :**

```
Enter the Order:- 2 3  
Enter the elements:-  
1 2 3  
3 4 5  
Your matrix is---  
1 2 3  
3 4 5
```

## **Experiment 7**

### **Programs using Dynamic memory allocation**

```
#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>  
  
int main()  
{  
    char *mem_allocation;  
    /* memory is allocated dynamically */  
    mem_allocation = malloc( 20 * sizeof(char) );  
    if( mem_allocation== NULL )  
    {  
        printf("Couldn't able to allocate requested memory\n");  
    }  
    else  
    {  
        strcpy( mem_allocation,"welcome to programming world");  
    }  
    printf("Dynamically allocated memory content : " \
```

```
        "%s\n", mem_allocation );  
    free(mem_allocation);  
}
```

### **OUTPUT :**

Dynamically allocated memory content : welcome to programming world

## **Experiment 8**

### **Programs using Templates and Exceptions**

#### **Program for designing a template function for implementing bubble sort.**

```
#include<iostream.h>  
#include<conio.h>  
#include<stdlib.h>  
template<class x,class y>void bubble(x *a,y n)  
{  
    x temp;  
    y i,j;  
    for(i=1;i<n;i++)  
    {  
        for(j=0;j<n-i;j++)  
        {
```

```

if(a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
}
void main()
{
int x[20],i,n,y;
float a[20];
clrscr();
do{
cout<<"\nPress 1 for integers or 2 for floats---";
cin>>y;
switch(y)
{
case 1:{
cout<<"\nEnter the range(for integers):- ";
cin>>n;
cout<<"\nEnter the numbers(for integers):- \n";
for(i=0;i<n;i++)
cin>>x[i];
bubble(x,n);
cout<<"\nAfter bubble sort ---\n";
for(i=0;i<n;i++)
cout<<x[i]<<" ";
break;
}
case 2:{
cout<<"\nEnter the range(for floats):- ";
cin>>n;
cout<<"\nEnter the numbers(for floats):- \n";
for(i=0;i<n;i++)
cin>>a[i];
bubble(a,n);
cout<<"\nAfter bubble sort ---\n";
for(i=0;i<n;i++)

```

```

cout<<a[i]<<" ";
break;
}
default: {
exit(0);
}
}
}
while(1);
getch();
}

```

## **OUTPUT :**

```

Press 1 for integers or 2 for floats--- 1
Enter the range(for integers):- 5
Enter the numbers(for integers):- 5 4 3 2 1
After bubble sort ---
1 2 3 4 5
Press 1 for integers or 2 for floats--- 2
Enter the range (for floats):- 5
Enter the numbers(for floats):- .5.1 4.1 3.1 2.1 1.1
After bubble sort ---
1.12.2 3.3 4.4 5.5

```

## **Experiment 9**

### **Programs using Sequential and Random access files**

```

#include<iostream.h>
#include<fstream.h>
#include<string.h>
#include<conio.h>
void main()
{
clrscr();

```

```
char str[100];
int len;
fstream deb;
deb.open("chintu.txt",ios::in|ios::out);
cout<<"Enter your string ---\n\n";
cin>>str;
len=strlen(str);
for(int i=0;i<len;i++)
{
if(str[i]==' ')
deb.put(' ');
else
deb.put(str[i]);
}
deb.seekg(0);
for(i=0;i<len;i++)
{
deb.get(str[i]);
cout<<str[i];
}
deb.close();
getch();
}
```