INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN

Object Oriented Programming
Practical File

BIT-204

Submitted by:

NAME :- Latika kardam
ROLL NO. :- 18901012020
CSE 2 B2

Submitted to:

Ms. VIBHA

INDEX

S.No.	Experiment Name	Date
1	Write a program to calculate the factorial of a number using functions.	14-01-21
2	Write a program to check if a number is prime or not using functions.	21-01-21
3	Write a program to calculate of a student using classes. It should maintain student's record. Input is basic info of the student and make member functions as public to display and read marks.	28-01-21
4	Write a program to show the use of i) Swapping using call by value, call by reference, call by address ii) Default value iii) and show function overloading.	04-02-21
5	Write a program to show the use of references, inline functions, macros, dynamic memory allocation using new and delete, scope and resolution operator.	11-02-21
6	Write a program to show addition of complex numbers using friend functions, objects as arguments by value and reference.	18-02-21
7	Write a program to perform operations on matrices using class like addition, subtraction, multiplication, transpose.	25-02-21
8	Write a program to calculate the area and perimeter of circle using classes.	25-02-21
9	Write a program to make a class for quadratic equation and find its roots.	4-03-21
10	WAP to add two timer (say t1,t2) using classes	
11	WAP to implement constructor & destructor, constructor overloading, copy constructor by making an employee class.	
12	WAP to show a comparison btw assignment operator & copy constructor.	

13	WAP to show different operator overloading.	
14	WAP to show all kinds of inheritance with constructors and scope resolution.	
15	WAP to show virtual function concept.	

Experiment No. 1

Aim:

Write a program to calculate the factorial of a number using functions.

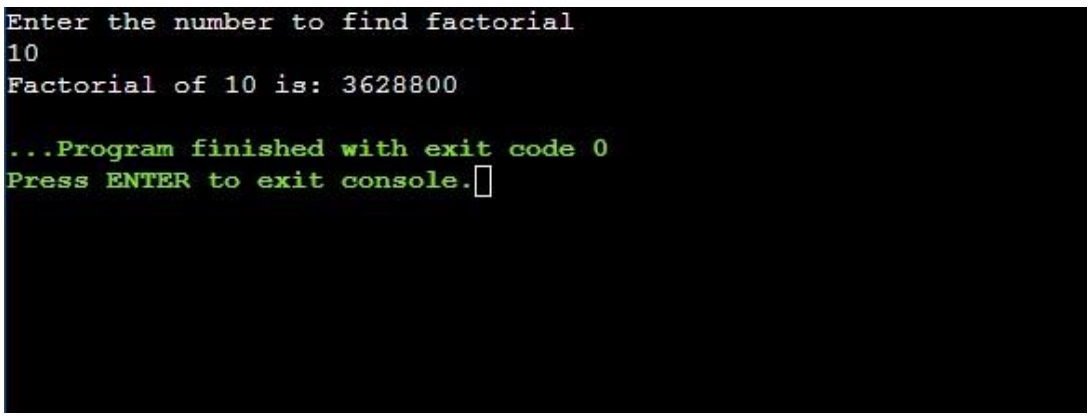
Code:

```
#include <iostream>
#include <conio.h>
using namespace std;

void factorial(int n)
{    int
fact=1,i;
   for(i=1; i<=n; i++)
   {
      fact=fact*i;
   }
   cout<<"Factorial of "<<n<<" is: "<<fact;
}

int
main()  {
int n;
   cout << "Enter the number to find factorial"
<< endl;    cin>>n;    factorial(n);    getch();
return 0;
}
```

Output:



Experiment No. 2

Aim:

Write a program to check if a number is prime or not using functions.

Code:

```
#include <iostream>
using namespace std;

void isPrime(int n)
{
  int i, flag = 0;
  for(i=2; i<=n/2; ++i)
  {
    if(n%i==0)
    {
flag=1;
break;
    }  }  if
(flag==0)
  cout<<n<<" is a prime
number"<<endl;  else
  cout<<n<<" is not a prime number"<<endl;
}      int
main()  {
int n;
   cout<<"Enter  a  number  to  check
prime  :  ";      cin>>n;      isPrime(n);
return 0;
}
```

Output:

```
Enter a number to check prime : 10
10 is not a prime number


...Program finished with exit code 0
Press ENTER to exit console.
```

```
Enter a number to check prime : 7
7 is a prime number


...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment No. 3

Aim:

Write a program to calculate of a student using classes. It should maintain student's record. Input is basic info of the student and make member functions as public to display and read marks.

Code:

```
#include
<iostream> using
namespace   std;
class marks
{   public:   float
f;   int
a,b,c,d,e,rollno;
  char name[70];

  void read()
  {
  cout << endl << "Enter the marks below:" << endl;
  cout << "Data Structures:
" ;   cin >> a;
  cout << "Discrete Structures:
" ;   cin >> b;
  cout << "Database Management:
" ;   cin >> c;
  cout << "Software engineering:
" ;   cin >> d;
  cout << "Material Science: " ;
  cin >> e;
  f =(a+b+c+d+e)/5;
  }

  void display()
  {
    cout<<"RollNo. :"<<rollno<<endl;
cout<<"Name   :"<<name<<endl;
    cout<<"Marks obtained in different
subjects:"<<endl;      cout<<"Data structures
:"<<a<<endl;      cout<<"Discrete structures
:"<<b<<endl;      cout<<"Database Management
:"<<c<<endl;      cout<<"Software
Engineering:"<<d<<endl;      cout<<"Material
Science   :"<<e<<endl;      cout<<"Percentage
obtained:"<<  f<<"%"<<endl;
```

Experiment No. 4

Aim:

}

```
};

int main ()
{
   int i,n,k,l;
marks pe[50];
   cout<< "Enter the total no. of students in the
class:";    cin>> n;    if (n>50)
   { cout<<"Invalid no. of students";}
   else {
   cout<<"----------Enter Student Details----------"<<endl;
for(i=0;i<n;i++)
   {
    cout<<"Rollno.:";
cin>>pe[i].rollno;
cout<<"Name   :" ;
cin>>pe[i].name;
cout<<endl;
   }
   for (k=0;k<n;k++)
   {
    cout<<"Rollno. "<< pe[k].rollno;
      pe[k].read();
      cout<<endl;
   }
   cout<<"----------Student Details----------
"<<endl;    for (l=0;l<n;l++)
   {
      pe[l].display();
      cout<<endl;
   }   }
return 0;
}
```

Aim:

Output:

```
Enter the total no. of students in the class:2
---------------Enter Student Details-----------------
Rollno.:158
Name   :ashlesha

Rollno.:189
Name   :latika

Rollno. 158
Enter the marks below:
Data Structures: 95
Discrete Structures: 89
Database Management: 97
Software engineering: 88
Material Science: 93

Rollno. 189
Enter the marks below:
Data Structures: 84
Discrete Structures: 63
Database Management: 86
Software engineering: 83
Material Science: 72

-----------------Student Details-----------------
RollNo. :158
Name    :ashlesha
Marks obtained in different subjects:
Data structures     :95
Discrete structures :89
Database Management :97
Software Engineering:88
Material Science    :93
Percentage obtained:92%

RollNo. :189
Name    :latika
Marks obtained in different subjects:
Data structures     :84
Discrete structures :63
Database Management :86
Software Engineering:83
Material Science    :72
Percentage obtained:77%



...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment No. 4

Write a program to show the use of i) Swapping using call by value, call by reference, call by address ii) Default value iii) and show function overloading.

Code:

```
#include <iostream>
using namespace std;

   int swap(int a, int b=0)
   {
int c;
c=a;
a=b;
b=c;
      cout<<"The number a and b respectively are: "<<a<<" and
"<<b<<endl;       return 0;
   }

   int swap(float *d, float *e)
{   float f;
f=*d;
*d=*e;
*e=f;
return 0;
}

int add(int *a, int *b)
{
   int          c;
c=(*a)+(*b);
return c;
}

int main()
{
   int num1, num2;
float num3, num4;
   cout<<endl<<"----------SWAPPING THE NUMBERS----------
"<<endl;    cout<<endl<<"Enter the integers to be swapped
below:"<<endl;    cout<<"a: ";    cin>>num1;    cout<<"b: ";
cin>>num2;
```

Aim:

```
   cout<<endl<<"Enter the real numbers to be swapped
below:"<<endl;    cout<<"d: ";    cin>>num3;
   cout<<"e: ";
cin>>num4;
   cout<<endl<<"----After Swapping----"<<endl;
   cout<<swap(&num3,&num4);
   cout<<"The number d and e respectively are: "<<num3<<" and
"<<num4<<endl;    cout<<endl<<swap(num1,num2);
   cout<<"(The numbers d and e respectively in the main function are:
"<<num3<<" and
"<<num4<<")"<<endl;
   cout<<"(The numbers a and b respectively in the main function are:
"<<num1<<" and
"<<num2<<")"<<endl;
   cout<<endl<<"-------------------------------------------------------------------------";
cout<<endl<< "Swapping after passing only one integer a in the
function:"<<endl;    cout<<swap(num1)<<endl;
   cout<<"(The numbers a and b respectively in the main function are:
"<<num1<<" and
"<<num2<<")"<<endl;
   cout<<endl<<"-------------------------------------------------------------------------";
cout<<endl<<"Now the addition of a and b is:
"<<add(&num1,&num2)<<endl<<endl;     cout<<endl<<"-------------------------
-----------------------------------------";    return 0;
}
```

Output:

Output

```
/tmp/jSKi0S3zYe.o
----------SWAPPING THE NUMBERS-----------

Enter the integers to be swapped below:
a: 1
b: 3
Enter the real numbers to be swapped below:
d: 3
e: 5
----After Swapping----
OThe number d and e respectively are: 5 and 3

The number a and b respectively are: 3 and 1
O(The numbers d and e respectively in the main function are: 5 and 3)
(The numbers a and b respectively in the main function are: 1 and 3)

--------------------------------------------------------------------------------
Swapping after passing only one integer a in the function:
The number a and b respectively are: 0 and 1
0
(The numbers a and b respectively in the main function are: 1 and 3)

--------------------------------------------------------------------------------
Now the addition of a and b is: 4


--------------------------------------------------------------------------------
```

Experiment No.

Aim:

5

Write a program to show the use of references, inline functions, macros, dynamic memory allocation using new and delete, scope and resolution operator.

Code:

```
#include <iostream>
using namespace std;

int s=40;

#define Print_square cout<<endl<<"Square of local variable s is: "<<s*s;

inline int sum (int a, int b)
{
   return a+b;
}

void increment (int& s)
{
s++;
}

int main ()
{  int
  a,b,c;
    cout<<endl<<"Enter the numbers to be added
  below: "<<endl;  cout<<"Enter a: ";  cin>>a;
  cout<<"Enter b: ";  cin>>b;   c=sum(a,b);
  cout<<endl<<"The sum of a and b is: "<<c;

  int s;
  s=12;
  cout<<endl<<"_____
  __
_____"<<endl;
  Print_square
  cout<<endl<<"s as local variable is: "<<s;
  increment(s);
  cout<<endl<<endl<<"-------After incrementing the value of s-------";
  Print_square
```

```
  cout<<endl<<endl<<"s as local variable is: "<<s;
  cout<<endl<<"s as global variable is: "<<::s;

  cout<<endl<<"_____
  __ _____"<<endl; int *d; d = new int; *d=10;
  cout<<endl<<"Pointer d points to a value of:
  "<<*d<<endl; delete d;
  cout<<endl<<"---------AFTER USING- delete operator---------";
  cout<<endl<<"Pointer d points to a value of: "<<*d<<endl<<endl;

  return 0;
}
```

Output:

```
Output
/tmp/WCWmnOvZMx.o
Enter the numbers to be added below:
Enter a: 5
Enter b: 10
The sum of a and b is: 15
_____

Square of local variable s is: 144
s as local variable is: 12

-------After incrementing the value of s-------
Square of local variable s is: 169

s as local variable is: 13
s as global variable is: 40
_____

Pointer d points to a value of: 10

---------AFTER USING- delete operator---------
Pointer d points to a value of: 0
```

Experiment No.

Aim:

6

Write a program to show addition of complex numbers using friend functions, objects as arguments by value and reference.

Code:

```
#include <iostream>
using namespace std;

class cl
{
private:
   int sum1, sum2;
float sumi, sumr;
public:
   int img1, img2, real1, real2;
   float img_1, img_2, real_1, real_2;

   friend void complex1(cl d);
friend int complex2(cl &d);
};

void complex1 (cl d)
{
   d.sum1 = d.img1 + d.img2;
   d.sum2 = d.real1 + d.real2;    cout<<endl<<endl<<"Sum of the
Two complex numbers in the function is:
"<<d.sum2<<" + i"<<d.sum1;
}

int complex2 (cl &d)
{
   d.sumi = d.img_1 + d.img_2;
   d.sumr = d.real_1 + d.real_2;    cout<<endl<<endl<<"Sum of
the Two complex numbers in the function is:
"<<d.sumr<<" + i"<<d.sumi;
}

int main()
{
   cl d;
```

```
   cout<<endl<<"Enter the complex numbers (int values) below:"<<endl;
cout<<endl<<"Complex Number 1:"<<endl;
   cout<<"Real part: ";
cin>>d.real1;
cout<<"Imaginary part: ";
cin>>d.img1;
   cout<<endl<<"Complex Number 2:"<<endl;
   cout<<"Real part: ";
cin>>d.real2;
cout<<"Imaginary part: ";
cin>>d.img2;
cout<<endl<<"The two complex numbers are: "<<d.real1<<" +
i"<<d.img1<<"  and
"<<d.real2<<" + i"<<d.img2;
   complex1(d);

cout<<endl<<"_____________________________________________________________
__
_____";
   cout<<endl<<endl<<"Enter the complex numbers (float values)
below:"<<endl;    cout<<endl<<"Complex Number 1:"<<endl;
   cout<<"Real part: ";
cin>>d.real_1;
cout<<"Imaginary part: ";
cin>>d.img_1;
   cout<<endl<<"Complex Number 2:"<<endl;
   cout<<"Real part: ";
cin>>d.real_2;
cout<<"Imaginary part: ";
cin>>d.img_2;
   cout<<endl<<"The two complex numbers are: "<<d.real_1<<" +
i"<<d.img_1<<"  and
"<<d.real_2<<" + i"<<d.img_2;
   complex2(d);

cout<<endl<<"_____________________________________________________________
_"
<<endl;
return 0;
}
```

Output:

Experiment No.

Aim:

```
Enter the complex numbers (int values) below:

Complex Number 1:
Real part: 3
Imaginary part: 2

Complex Number 2:
Real part: 5
Imaginary part: 1

The two complex numbers are: 3 + i2  and  5 + i1

Sum of the Two complex numbers in the function is:     8 + i3
_____________________________________________________________

Enter the complex numbers (float values) below:

Complex Number 1:
Real part: 2.1
Imaginary part: 5.4

Complex Number 2:
Real part: 1.1
Imaginary part: 9.7

The two complex numbers are: 2.1 + i5.4  and  1.1 + i9.7

Sum of the Two complex numbers in the function is: 3.2 + i15.1
_____________________________________________________________


...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment No.

7

Aim: **Write a program to perform operations on matrices using class like addition, subtraction, multiplication, transpose.**

Code:

```
#include <iostream>
using namespace std;

class arr {      public:
int              a[20][20],
b[20][20];
};

void trans(int a[][20], int m, int n)
{
   int b[20][20];    for
(int i=0; i<m; i++)
   {
      for (int j=0; j<n; j++)
      {
         b[j][i]=a[i][j];
      }
   }
   cout<<endl<<"The transpose: A'"<<endl;
   for(int i=0; i<n; i++)
   {
      for (int j=0; j<m; j++)
      {
         cout<<b[i][j]<<" ";
      }
      cout<<endl;
   } }

void mul(int a[][20], int b[][20], int m, int n, int q)
{  int
c[20][20];
 cout<<endl<<"The product: A*B"<<endl;
 for (int i=0; i<m; i++)
 {
    for (int j=0; j<q; j++)
    {
       c[i][j] =0;
       {
```

```
        for (int k=0; k<n; k++)
        {
        c[i][j]=c[i][j]+(a[i][k]*b[k][j]);
        }
       }
       cout<<c[i][j]<<" ";
    }
    cout<<endl;
 }
}

void add( int a[][20], int b[][20], int m, int n)
{   int
c[20][20];
   for (int i=0; i<m; i++)
   {
      for (int j=0; j<n; j++)
      {
         c[i][j]=b[i][j]+a[i][j];
      }
   }
   cout<<endl<<"The sum: A+B"<<endl;
for(int i=0; i<m; i++)
   {
      for (int j=0; j<n; j++)
      {
         cout<<c[i][j]<<" ";
      }
      cout<<endl;
   }
}

void sub(int a[][20], int b[][20], int m, int n)
{       int c[20][20];
for (int i=0; i<m; i++)
   {
      for (int j=0; j<n; j++)
      {
         c[i][j]=a[i][j]-b[i][j];
      }
   }
   cout<<endl<<"The difference: A-B"<<endl;
   for(int i=0; i<m; i++)
   {
      for (int j=0; j<n; j++)
      {
```

```
        cout<<c[i][j]<<" ";
      }
      cout<<endl;
   }
}

int main()
{           int
  m,n,p,q;
  arr dr;
   cout<<endl<<"Enter the number of rows in matrix A:
  ";  cin>>m;
   cout<<"Enter the number of columns in matrix A:
  ";  cin>>n;
   cout<<endl<<"Enter the number of rows in matrix B:
  ";  cin>>p;
   cout<<"Enter the number of columns in matrix B:
  ";  cin>>q;
   cout<<endl<<"Enter the elements of matrix A
  below:"<<endl;  for(int i=0; i<m; i++)
   {
      for (int j=0; j<n; j++)
      {
        cout<<"A["<<i+1<<"]["<<j+1<<"]: ";
  cin>>dr.a[i][j];
      }
   }
   cout<<endl<<"Enter the elements of matrix B
  below:"<<endl;  for(int i=0; i<p; i++)
   {
      for (int j=0; j<q; j++)
      {
        cout<<"B["<<i+1<<"]["<<j+1<<"]: ";
  cin>>dr.b[i][j];
      }
   }
  cout<<endl<<"_____Matrices A and
  B_____"<<endl; cout<<endl<<" A: "<<endl;    for(int i=0;
  i<m; i++)
     {
        for (int j=0; j<n; j++)
        {
           cout<<dr.a[i][j]<<" ";
        }
        cout<<endl;
     }
     cout<<endl<<" B: "<<endl;
```

```
    for(int i=0; i<p; i++)
    {
       for (int j=0; j<q; j++)
       {
          cout<<dr.b[i][j]<<" ";
       }
       cout<<endl;
    }
   cout<<endl<<"_____Addition of matrices A +
  B_____"<<endl;
   if (m==p&&n==q)
   {
      add(dr.a,dr.b,m,n);
   }
  else
  {
      cout<<endl<<"Addition of matrices is not possible!!!!";
   }

   cout<<endl<<endl<<"_____Subtraction of matrices A -
  B_____"<<endl;
   if (m==p&&n==q)
   {
      sub(dr.a,dr.b,m,n);
   }
  else
  {
      cout<<endl<<"Subtraction of matrices is not possible!!!!";
   }
    cout<<endl<<endl<<"_____Multiplication of matrices
  A*B_____"<<endl;
   if (n==p)
  {
      mul(dr.a,dr.b,m,n,q);
   }
  else
  {
      cout<<endl<<"Multiplication of matrices is not possible!!!!";
   }
    cout<<endl<<endl<<"_____Transpose of matrix
  A_____"<<endl;
  trans(dr.a,m,n);
  cout<<endl<<endl;
   return 0;
}
```

Output:

```
Enter the number of rows in matrix A: 3
Enter the number of columns in matrix A: 3

Enter the number of rows in matrix B: 3
Enter the number of columns in matrix B: 3

Enter the elements of matrix A below:
A[1][1]: 2
A[1][2]: 1
A[1][3]: 5
A[2][1]: 3
A[2][2]: 7
A[2][3]: 9
A[3][1]: 4
A[3][2]: 0
A[3][3]: 6

Enter the elements of matrix B below:
B[1][1]: 0
B[1][2]: 2
B[1][3]: 9
B[2][1]: 5
B[2][2]: 4
B[2][3]: 6
B[3][1]: 7
B[3][2]: 1
B[3][3]: 1

_____Matrices A and B_____

 A:
2 1 5
3 7 9
4 0 6

 B:
0 2 9
5 4 6
7 1 1

_____Addition of matrices A + B_____

The sum: A+B
2 3 14
8 11 15
11 1 7


_____Subtraction of matrices A - B_____

The difference: A-B
2 -1 -4
-2 3 3
-3 -1 5


_____Multiplication of matrices A*B_____

The product: A*B
40 13 29
98 43 78
42 14 42


_____Transpose of matrix A_____

The transpose: A'
2 3 4
1 7 0
5 9 6




...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment No. 8

Aim: Write a program to calculate the area and perimeter of circle using classes.

Code:

```
#include<iostrea
m>
#include<conio.h
>                using
namespace   std;
class Circle
{
private:
float r;
public:
   void input()
   {
      cout<<"Enter radius of circle: ";
      cin>>r;
   }
   void Area()
   {
      float a=22.0/7.0*r*r;
      cout<<"\nArea of circle ="<<a;
   }
   void Circumference()
   {
      float p=2*22.0/7.0*r;
      cout<<"\nCircumference of circle ="<<p;
   }  };  int main()  {
Circle              c1;
c1.input();
c1.Area();
c1.Circumference
();
   return 0;
}
```

Output:

```
Enter radius of circle: 10

Area of circle =314.286
Circumference of circle =62.8571

...Program finished with exit code 0
Press ENTER to exit console.
```

Experiment No. 9

Aim: Write a program to make a class for quadratic equation and find its roots.

Code:

```
#include
<iostream>
#include <cmath>
using namespace
std;

class roots {
int  a,  b,  c;
float  r1,  r2;
public:
      void getdata()
      {
          cout << "Enter value of coefficient of
x^2: ";          cin >> a;
          cout << "Enter value of coefficient of
x: ";            cin >> b;
          cout << "Enter value of coefficient of
1: ";            cin >> c;
      }
    int determinant()
      {
        int  d  =  b  *  b;
d  -=  (4  *  a  *  c);
return d;
      }
  void checkdeterminant(int d)
      {

          if (d == 0)
```

```
        {
           cout << "Real and equal
roots\n";                       r1 = (-1 *
b);                     r1 /= (2 * a);
          r2 = r1;
               cout << "Roots : " << r1 << " and " << r2 << endl;
        }
         else if (d > 0)
        {
                    cout << "Real and distinct
roots\n";                       r1 = (-1 * b) +
sqrt(d);                r1 /= (2 * a);
r2 = (-1 * b) - sqrt(d);                r2 /= (2 *
a);
               cout << "Roots : " << r1 << " and " << r2 << endl;
          }
else
        {
       cout << "Imaginary roots" << endl << endl;
r1 = (-b)/(2*a);        r2 = (sqrt(-d))/(2*a);
                    cout << "Roots : " << r1 << " + i" << r2 ;
          cout << " and " << r1 << " - i" << r2 << endl<<endl;
        }
     }
};

int main()
{
   roots r;

 r.getdata();   cout  <<
endl;        int    d    =
r.determinant();
r.checkdeterminant(d
);
   cout << endl;

   return 0;
}
```

Output:

Output

```
/tmp/RRlamJx9Y6.o
Enter value of coefficient of x^2: 5
Enter value of coefficient of x: -2
Enter value of coefficient of 1: -9
Real and distinct roots
Roots : 1.55647 and -1.15647
```