LAB EXPERIMENTS (OF OOPS WITH C++)

1] Develop a C++ program to find the largest of three numbers.

```
#include <iostream>   //HEADER FILES
using namespace std;
int main()
{
   int a,  b,  c;       //DECLARING THREE LOCAL VARIABLES
   cout<<"ENTER THREE VALUES:";
   cin>>a>>b>>c;      //READS THREE VALUES SIMULTANEOUSLY
        If (a>b && a>c)      //ENTERS THE IF STATEMENT
              cout<<"THE GREATEST IS :"<<a;
         if (b>a && b>c)
              cout<<"THE GREATEST IS :"<<b;
          else
               cout<<"THE GREATEST IS :"<<c;
   return 0;
}
```

OUTPUT:

```
ENTER THREE VALUES   : 2
3
4
THE GREATEST IS   : 4
```

2] Develop a C++ program to sort the elements in ascending and descending order.

```
#include <iostream>
using namespace std;
int main()
{
	int num [10], n;
int i, j, man;
cout<<"enter n for the numbers you want to sort"<<endl;
cin>>n;
	for(i=0;i<n;i++)
{
		cout<<"enter number :"<<endl;
	cin>>num[i];
	}
	for(i=0;i<n;i++)
{
		for(j=0;j<n;j++)
{
			if(num[i]<num[j])
{
				man=num[i];
				num[i]=num[j];
				num[j]=man;
```

```
        }
        }
        }
        cout<<"ascending "<<endl;
        for(i=0;i<n;i++)
{
        cout<<" "<<num[i]<<endl;
}
        for(i=0;i<n;i++)
{
                for(j=0;j<n;j++)
{
                        if(num[i]>num[j])
{
                                man=num[i];
                                num[i]=num[j];
                                num[j]=man;
                }
                }
        }
        cout<<" descending"<<endl;
        for(i=0;i<n;i++){
                cout<<" "<<num[i]<<endl;
}
```

```
return 0;
}
```

OUTPUT:

```
enter n for the numbers you want to sort
3
enter number :
1
enter number :
2
enter number :
3
ascending
 1
 2
 3
 descending
 3
 2
 1
```

3] Develop a C++ program using classes to display student name, roll number, marks obtained in two subjects and total score of the student.

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

```
class Student
 {
    public:
  string name;
  int rno;
  int mrks1;
  int mrks2;
  int total_score()
 {
    return mrks1 + mrks2;
  }
};

int main()
{
  Student s1;
  cout << "Enter the name of the student: ";
  cin >> s1.name;
  cout << "Enter the roll number of the student: ";
  cin >> s1.rno;
  cout << "Enter the marks of the student in subject 1: ";
  cin >> s1.mrks1;
  cout << "Enter the marks of the student in subject 2: ";
```

```
  cin >> s1.mrks2;
cout << "The name of the student is: " << s1.name << endl;
  cout << "The roll number of the student is: " << s1.rno << endl;
  cout << "The marks of the student in subject 1 are: " << s1.mrks1 << endl;
  cout << "The marks of the student in subject 2 are: " << s1.mrks2 << endl;
  cout << "The total score of the student is: " << s1.total_score() << endl;
return 0;
}
```

OUTPUT:

```
Enter the name of the student: XYZ
Enter the roll number of the student: 80
Enter the marks of the student in subject 1: 100
Enter the marks of the student in subject 2: 87
The name of the student is: XYZ
The roll number of the student is: 80
The marks of the student in subject 1 are: 100
The marks of the student in subject 2 are: 87
The total score of the student is: 187
```

4] Develop a C++ program for a bank empolyee to print name of the employee, account_no. & balance.

Print invalid balance if amount<500, Display the same, also display the balance after the withdraw and deposit.

```
#include<iostream>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
using namespace std;
class bank
{
      int acno;
      char nm[100], acctype[100];
      float bal;
  public:
      bank(int acc_no, char *name, char *acc_type, float balance)
//Parameterized Constructor
      {
            acno=acc_no;
            strcpy(nm, name);
            strcpy(acctype, acc_type);
            bal=balance;
      }
      void deposit();
      void withdraw();
      void display();
};
void bank::deposit()   //depositing  amount
```

```
{
	int damt1;
	cout<<"\n Enter Deposit Amount = ";
	cin>>damt1;
	bal+=damt1;
}
void bank::withdraw()  //withdrawing  amount
{
	int wamt1;
	cout<<"\n Enter Withdraw Amount = ";
	cin>>wamt1;
	if(wamt1>bal)
		cout<<"\n Cannot Withdraw Amount";
	bal-=wamt1;
}
void bank::display()  //displaying the details
{
	cout<<"\n ----------------------";
	cout<<"\n Accout No. : "<<acno;
	cout<<"\n Name : "<<nm;
	cout<<"\n Account Type : "<<acctype;
	cout<<"\n Balance : "<<bal;
}
int main()
```

```
{
        int acc_no;
        char name[100], acc_type[100];
        float balance;
        cout<<"\n Enter Details: \n";
        cout<<"----------------------";
        cout<<"\n Accout No. ";
        cin>>acc_no;
        cout<<"\n Name : ";
        cin>>name;
        cout<<"\n Account Type : ";
        cin>>acc_type;
        cout<<"\n Balance : ";
        cin>>balance;

 if (balance<=500)
 {
cout<<"\n Invalid";
 }
 else
 {
  bank b1(acc_no, name, acc_type, balance);  //object is created
        b1.deposit(); //
        b1.withdraw(); // calling member functions
```

```
        b1.display(); //
 }
 return 0;
}
```

OUTPUT:

```
Enter Details:
----------------------
 Accout No. 2745
 Name : XDC
 Account Type : Savings
 Balance : 9000
 Enter Deposit Amount = 1000
 Enter Withdraw Amount = 2000
 ----------------------
 Accout No. : 2745
 Name : XDC
 Account Type : Savings
 Balance : 8000
```

2nd OUTPUT:

```
Enter Details:
----------------------
 Accout No. 9190
 Name : ZYX
 Account Type : Savings
```

Balance : 20

Invalid

5] Develop a C++ program to demonstrate function overloading for the following prototypes.

add(int a, int b)

add(double a, double b) .

```
#include <iostream>
using namespace std;
void add(int a, int b)
{
  cout << "sum = " << (a + b);
}
 void add(double a, double b)
{
    cout << endl << "sum = " << (a + b);
}
 int main()
{
    add(10, 2);
    add(5.3, 6.2);
    return 0;
}
```

OUTPUT:

sum = 12

sum = 11.5

6] Develop a C++ program using Operator Overloading for overloading Unary minus operator.

```
#include<iostream>
using namespace std;
 class NUM
{
   private:
      int n;
   public:
      void getNum(int x)     //function to get number
      {
         n=x;
      }
      void dispNum(void)      //function to display number
      {
         cout << "value of n is: " << n;
      }
      //unary - operator overloading
      void operator - (void)
      {
         n=-n;
      }
};
```

```
int main()
{
   NUM num;
   num.getNum(10);
   -num;
   num.dispNum();
   cout << endl;
   return 0;
}
```

OUTPUT: value of n is: -10 .

7] Develop a C++ program to implement Multiple inheritance for performing arithmetic operation of two numbers.

```
#include <iostream>
using namespace std;
class base1
{
private:
      int a ,b, c;
        public:
         void input()
   {
      cout<<"ENTER TWO NUMBERS FOR THE SUM :";
      cin>>a>>b;
   }
```

```
void show()
{
   c=a+b;
   cout<<"SUM="<<c<<endl;
} };
class base2
{   private:
    int a,b,c;
    public:
      void input1()
      {
          cout<<"ENTER TWO NUMBERS FOR THE DIFFERENCE :";
          cin>>a>>b;
   }
   void show1()
{
   c=a-b;
   cout<<"DIFFERENCE ="<<c<<endl;
} };
class derive:public base1,public base2
{
   private:
   int a,b,c;
   public:
```

```
        void input2()
       {
       cout<<"ENTER TWO NUMBERS FOR THE MULTIPLICATION :";
           cin>>a>>b;
}
void show2()
{
   c=a*b;
   cout<<"MULTIPLICATION ="<<c<<endl;
}  };
int main()
{
   derive ob2;
   ob2.input2();
   ob2.show2();
   ob2.input();
   ob2.show();
   ob2.input1();
   ob2.show1();
   return 0;
}
```

OUTPUT:

```
ENTER TWO NUMBERS FOR THE MULTIPLICATION :2
3
```

MULTIPLICATION =6

ENTER TWO NUMBERS FOR THE SUM :78

90

SUM=168

ENTER TWO NUMBERS FOR THE DIFFERENCE :101

79

DIFFERENCE =22

8] Develop a C++ program using Constructor in Derived classes to initialize alpha, beta and gamma and display corresponding values.

```
#include <iostream>
using namespace std;
class Base
 {
    public:
  int alpha,beta;
   Base(int alpha, int beta)
{
    this->alpha = alpha;
    this->beta = beta;
  }     };
class Derived : public Base
{
public:
  int gamma;
```

```
 Derived(int alpha, int beta, int gamma) : Base(alpha, beta)
 {
    this->gamma = gamma;
  }
 void display()
{
    cout << "alpha = " << alpha << endl;
    cout << "beta = " << beta << endl;
    cout << "gamma = " << gamma << endl;
  }
};
int main()
{
  int alpha, beta, gamma;
  cout << "Enter the value of alpha: ";
  cin >> alpha;
   cout << "Enter the value of beta: ";
  cin >> beta;
  cout << "Enter the value of gamma: ";
  cin >> gamma;
 Derived derived(alpha, beta, gamma);
 derived.display();
return 0;   }
```

OUTPUT:

Enter the value of alpha: 12

Enter the value of beta: 34

Enter the value of gamma: 70

alpha = 12

beta = 34

gamma = 70