1. Write a program using two classes named A and B where the constructor of each class takes one integer and one-character value as argument and the constructor shows the message "In A class", "In B class" with the values. Complete the program considering the following case:

   i) A class inherits B class.

   Create an object of A class and show the output.

2. A class named Account contains an acc_no and balance. When anyone creates a new account, he has to set his new acc_no and balance. An account holder's balance can be known automatically only by his Son (class). But, Son's child (class) will not see the balance automatically. Design this scenario in code using OOP concepts. In main function write appropriate code to show your code works according to given constraints.

Question 1:

```cpp
#include<iostream>
using namespace std;
class B{
    public:
        B(int a, char b){
            cout << "In B Class" << endl;
        }

};
class A : public B{
    public:
        A(int a, int b) : B(a,b){
            cout << "In A Class" << endl;
        }

};
int main(){
    A ob(10,'a');
}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

TERMINAL
PS C:\Users\coder\Desktop\Class work c++> .\main.exe
In B Class
In A Class
```

Question 2:

```cpp
#include<iostream>
using namespace std;
class Account{
    int acc_no;
    public:
        int balance;
        Account(int a, int b){
            acc_no = a;
            balance = b;
        }
};
class child : private Account{
        public:
            child(int a, int b): Account(a,b){

            }
            void showBal(){
                cout << balance << endl;
            }
};
class childSon : public child{
      public:
         childSon(int a, int b): child(a,b){

         }
};
int main(){
    childSon ob(11101111,113333);
    ob.showBal();
}
```

```
PS C:\Users\coder\Desktop\Class work c++> .\main.exe
113333
```

CT-02

1. Write a program using three classes named alpha, beta, gamma where the constructor of each class takes one integer and one-character value as argument and the constructor contains the message "In alpha class", "In beta class" and in "In gamma class" respectively. Complete the program considering the following cases:
   i) beta inherits alpha and gamma inherits beta.
   Now show the output for both cases.

2. Give an example of multiple inheritance

Question 1:

```cpp
#include<iostream>
using namespace std;
class alpha{
    public:
        alpha(int x, char y){
            cout << "In alpha class" << endl;
        }
};

class beta:public alpha{
    public:
        beta(int x, char y):alpha(x, y){
            cout << "In beta class" << endl;
        }
};

class gamma:public beta{
    public:
        gamma(int x, char y):beta(x, y){
            cout << "In gamma class" << endl;
        }
};

int main(){
    gamma g(12,'a');
}
```

```
In alpha class
In beta class
In gamma class
```

Question 2:

```cpp
#include<iostream>
using namespace std;
class A{
    public:
        int x;
};

class B{
    public:
        int y;
};

class C: public A, public B{
    public:
        int z;
};

int main(){
    C ob;
    ob.x = 12;
    ob.y = 13;
    ob.z = 22;
    cout << ob.x << " " << ob.y << " " << ob.z << endl;
}
```

```
PS C:\Users\coder\Desktop\Class work c++> .\main.exe
12 13 22
```

1. Write a generic function PerfectNumber() which will take one argument and determine whether the argument is perfect or not. The method is of generic type.

2. Define a class **Merge_string** having two member variables - length (denoting string length) and s (denoting a string content) and member function display(). Overload + operator for concatenating two strings.

```
main(){
  Merge_string o1("Good "), o2("Morning"), o3;
  o3= o1+o2;
  o3.display();
}
```

Output:

Good Morning

3. Create an class named Shop. Shop will have a pure virtual function show() which will output what it sells. Create a derived class Book and Rice. Bookstore will define show() so that it will output "sells books" and similarly Rice will output "sells rice". In the main , use Shop class in a way that it becomes Book and Rice and call show() function.

Question 1:

```cpp
#include<iostream>
using namespace std;
template<typename T>
void PerFectNumber(T n){
    int x = 1;
    for(int i = 2; i < n; i++){
        if(n % i == 0){
            x += i;
        }
    }
    if(x == n){
        cout << "Perfect Number" << endl;
    }else{
        cout << "Not a perfect number" << endl;
    }
}

int main(){
    PerFectNumber(6);
    PerFectNumber(28);
    PerFectNumber(13);
}
```

```
PS C:\Users\coder\Desktop\Class work c++> .\main.exe
Perfect Number
Perfect Number
Not a perfect number
```

Question 2:

```cpp
#include<iostream>
using namespace std;
class Merge_string{
    public:
        string a;
        Merge_string(){
            a = " ";
        }
        Merge_string(string s){
            a = s;
        }

        Merge_string operator +(Merge_string s1){
                Merge_string ob;
                ob.a = a + s1.a;

                return ob;
        }

        void display(){
            cout << a << endl;
        }

};

int main(){
    Merge_string o1("Good "), o2("Morning"), o3;
    o3 = o1 + o2;
    o3.display();
}
```

```
PS C:\Users\coder\Desktop\Class work c++> .\main.exe
Good Morning
```

Question 3:

```cpp
#include<iostream>
using namespace std;
class Shop{
    public:
        virtual void show()=0;
};
class Book : public Shop{
    public:
        void show(){
            cout << "sells books" << endl;
        }
};
class Rice : public Shop{
    public:
        void show(){
            cout << "sells rice" << endl;
        }
};

int main(){
    Shop * shop;
    Book book;
    Rice rice;

    shop = &book;
    shop->show();

    shop = &rice;
    shop->show();
}
```

```
PS C:\Users\coder\Desktop\Class work c++> .\main.exe
sells books
sells rice
```

1. Create an class named Shop. Shop will have a pure virtual function show() which will output what it sells. Create a derived class Book and Rice. Bookstore will define show() so that it will output "sells books" and similarly Rice will output "sells rice". In the main , use Store class in a way that it becomes Bookstore and Rice and call show() function.

2.

```
main(){
FinalExam e1(40), e2(30),e3(70),e4;
e4=e1+e2;
if(e3 = = e4){
cout<<"equal";
}
else{
cout<<"Not equal";
} }
```

Complete this code to compare objects.

3. Write a program to compare three values and return the greatest one. The value can be integer, character or floating point numbers. You can write only one function to implement this.

Question 1:

```cpp
#include<iostream>
using namespace std;
class Shop{
    public:
        virtual void show()=0;
};
class Book : public Shop{
    public:
        void show(){
            cout << "sells books" << endl;
        }
};
class Rice : public Shop{
    public:
        void show(){
            cout << "sells rice" << endl;
        }
};

int main(){
    Shop * shop;
    Book book;
    Rice rice;

    shop = &book;
    shop->show();

    shop = &rice;
    shop->show();
}
```

```
PS C:\Users\coder\Desktop\Class work c++> .\main.exe
sells books
sells rice
```

Question 2:

```cpp
#include<iostream>
using namespace std;
class FinalExam{
    public:
        int a;
        FinalExam(){
            a = 0;
        }
        FinalExam(int s){
            a = s;
        }
        FinalExam operator +(FinalExam s1){
                FinalExam ob;
                ob.a = a + s1.a;

                return ob;
        }
        bool operator ==(FinalExam s1){
                if(s1.a == a){
                    return true;
                }else{
                    return 0;
                }
        }
};
int main(){
    FinalExam e1(40), e2(30), e3(70), e4;
    e4 = e1 + e2;

    if(e3 == e4){
        cout << "Equal";
    }else{
        cout << "Not Equal";
    }
}
```

```
PS C:\Users\coder\Desktop\Class work c++> .\main.exe
Equal
```

Question 3:

```cpp
#include<iostream>
using namespace std;
template <typename T>
T MaxNum(T n1, T n2, T n3){
  /*Find max number with conditional Operator
    T max = x > y ? x > z ?  x  :  z : y > z?  y :  z;
    return max;
  */

  if (n1 >= n2 && n1 >= n3){
    return n1;
  }
  else if (n2 >= n1 && n2 >= n3){
    return n2;
  }
  else{
    return n3;
  }

}
int main(){
    cout << MaxNum(18,3,18) << endl;
    cout << MaxNum('a','k','z') << endl;
    cout << MaxNum(1.001,1.003,1.002) << endl;
}
```

```
18
z
1.003
PS C:\Users\coder\Desktop\Class work C++>
```