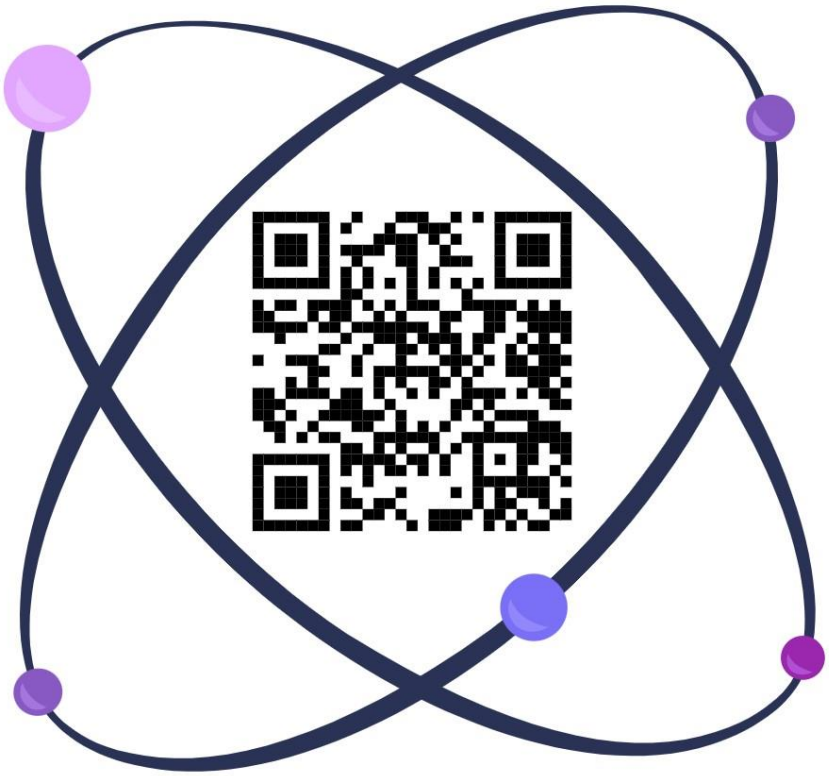
INDEX



1. **Write a program to enter marks of 6 different subjects and find out the total marks (Using cin and cout statement).**

```
#include<iostream>
using namespace std; int
main()
{
    float sub1, sub2, sub3, sub4, sub5, sub6;

    cout<<"Enter the marks of Math :";
cin>>sub1;
    cout<<"Enter the marks of OOPs :";
cin>>sub2;
    cout<<"Enter the marks of FOS :";
cin>>sub3;
    cout<<"Enter the marks of CSA :";
cin>>sub4;
    cout<<"Enter the marks of EVS :";
cin>>sub5;
    cout<<"Enter the marks of English :";
cin>>sub6;

    float totalMarks=sub1+sub2+sub3+sub4+sub5+sub6;

    cout<<"Total marks of the subject :"<<totalMarks<<endl;

    float average=totalMarks/6;
    cout<<"Presentage of all subject :"<<average<<"%";
return 0;
}
```

OUTPUT :-

```
Enter the marks of Math :46
Enter the marks of OOPs :45
Enter the marks of FOS :42
Enter the marks of CSA :44
Enter the marks of EVS :41
Enter the marks of English :40
Total marks of the subject :258
Presentage of all subject :43%
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

2.write a function using reference variable as argument to swap the values of pair of integers.

```
#include<iostream>
using namespace std;
int main() {
    int a, b, swap;
    cout<<"Enter the first number :";
cin>>a;
    cout<<"Enter the second number :";
cin>>b;

    swap = a;
a = b;     b
= swap;
    cout<<"\nResult"<<endl;
    cout<<"Swaping number of A ="<<a<<endl;
cout<<"Swaping number of B ="<<b;

    return 0;

}
```

OUTPUT :-

```
Enter the first number :22
Enter the second number :10

Result
Swaping number of A =10
Swaping number of B =22
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

3. Write a function to find largest of three number.

```
#include<iostream>
using namespace std;
 int
main() {
    int a, b, c;

    cout<<"Enter the first number :";
cin>>a;
    cout<<"Enter the second number :";
cin>>b;
    cout<<"Enter the third numbrer :";
cin>>c;
    cout<<"Show result"<<endl;

    if(a>b&&a>c)
{
        cout<<"A is the largest Number:"<<a;
    }
    if(b>a&&b>c)
{
        cout<<"B is the largest Number:"<<b;
    }
else
{
        cout<<"C is the largest number:"<<c;
```

```
    }
return 0;

}
```

OUTPUT :-

```
Enter the first number :22
Enter the second number :10
Enter the third numbrer :32
Show result
C is the largest number:32
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

4. Write a program to find the factorial of a number.

```
#include<iostream>
using namespace std;
 int main() {
int num, fact;
fact = 1;

    cout<<"Enter any Number :";
cin>>num;

    for(int i=1; i<=num; i++)
    {
        fact = fact*i;
    }
    cout<<"\nResult"<<endl;
cout<<"Factorial is :"<<fact;     return
0;
}
```

OUTPUT :-

```
Enter any Number :12

Result
Factorial is :479001600
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

5.Define a class to represent a bank account which includes the following members as data members:

A) Name of the depositor, B) Account Number, C) Withdrawal amount, D) Balance amount in the account

A) To assign initial values, B) To deposit an amount, C) To withdraw an amount after checking the balance, D) To display name and balance

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

class bank_account_details {
private:     string name;
int ac_number;     int
amount;     int w_amount;
double balance = 0;
 public:     void
assign_initial_value()
    {
        cout << "Enter the depositor name :";
cin >> name;
        cout << "Enter account number :";
        cin >> ac_number;         cout
<< "Enter your balance :";
        cin >> balance;
    }
    void deposit() {
        cout << "Enter amount you want to withdraw :";
cin >> w_amount;         if (w_amount <= balance)
        {
            balance = balance - w_amount;
            cout << "\nYour amount is successfully withdrawn.";
cout << "Your available balance is :" << balance << endl;
        } else {
            cout << "You have insufficient balance.";
        }
    }
    void display()
    {
        cout << "Account holder name :" << name << endl;
```

```
        cout << "Account number :" << ac_number << endl;
cout << "Balance: " << balance;
    }
};
 int
main()  {
    bank_account_details ad;
ad.assign_initial_value();
ad.deposit();
ad.display();     return 0;
}
```

Output :-

```
Enter the depositor name :Vinod
Enter account number :1234567890
Enter your balance :10000
Enter amount you want to withdraw :3000

Your amount is successfully withdrawn.Your available balance is :7000
Account holder name :Vinod
Account number :1234567890
Balance: 7000
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

6. **Write the above program for handling n number of account holders using array of objects.**

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

class bank_account_details
{   private:      string
depositor_name;      int
ac_number;      int
d_amount;      int
```

```
w_amount;      double
balance=0;
   public:      void
assign_initial_value()
     {
        cout<<"Enter your name :";
cin>>depositor_name;
        cout<<"Enter your account number :";
        cin>>ac_number;
cout<<"Enter your balance :";
cin>>balance;
     }
     void deposit()
     {
        cout<<"Enter amount you want to deposit :";
cin>>d_amount;         balance = balance +
d_amount;
        cout<<"\nYour amount is Successfully deposit.\nYour available
balance is :"<<balance<<endl;
     }
     void withdrow()
     {
        cout<<"\nEnter amount you want to withdrow :";
cin>>w_amount;

        if(w_amount <= balance)
        {
            balance = balance - w_amount;
cout<<"\nYour amount is Successfully withdrow.\nYour
balance is :"<<balance<<endl;
        }
```

```
        else
{
            cout<<"You have unsufficient balance\n\n";
        }
     }
     void display()
     {
        cout<<"\nAccount Holder's name
:"<<depositor_name<<endl;
        cout<<"Account number :"<<ac_number<<endl;
cout<<"Balance :" <<balance<<endl;
     }
};
 int
main() {
int i;
    bank_account_details c[10];

    for(int i=1;i<2;i++)
    {
        c[i].assign_initial_value();
c[i].deposit();         c[i].withdrow();
c[i].display();
    }     cout<<"\n\n";
bank_account_details clint;
clint.assign_initial_value();
clint.deposit();
clint.withdrow();
clint.display();

    return 0;
}
```

OUTPUT :-

```
Enter your name :Vinod
Enter your account number :1234567890
Enter your balance :10000
Enter amount you want to deposit :12000

Your amount is Successfully deposit.
Your available balance is :22000

Enter amount you want to withdrow :5000

Your amount is Successfully withdrow.
Your balance is :17000

Account Holder's name :Vinod
Account number :1234567890
Balance :17000


Enter your name :Ravikishan
Enter your account number :1234567890
Enter your balance :16000
Enter amount you want to deposit :5000

Your amount is Successfully deposit.
Your available balance is :21000

Enter amount you want to withdrow :1000

Your amount is Successfully withdrow.
Your balance is :20000

Account Holder's name :Ravikishan
Account number :1234567890
Balance :20000
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

7. **Write a C++ program to compute area of right angle triangle, equilateral triangle, isosceles triangle using function overloading concept.**

```
 #include<iostream>
using namespace std;
class Area   {
public:     float ar;

    void area(int b, int h)
    {
ar=(b*h)/2;
      cout<<"\nArea of right angle is :"<<ar;
    }
    void area(float a, float b, float c)
    {
      if(a==b && b==c && a==c)
      {
        cout<<"\n\nTriangle is equilater.";
ar=(sqrt(3)/4*(a*a));         cout<<"\nArea
is :"<<ar;
      }
      if(a==b || b==c || a==c)
      {
        cout<<"\n\nTriangle is isosceles.";
ar=b*sqrt(4*(a*a))-(b*b)/4;         cout<<"\nArea
is :"<<ar;
      }
    } };
int main()
{
    cout<<"Result!";
Area obj;
obj.area(6,10);
obj.area(4,4,4);
obj.area(2,2,10);
return 0;
  }
```

OUTPUT :-

```
Result!
Area of right angle is :30

Triangle is equilater.
Area is :6.9282

Triangle is isosceles.
Area is :28

Triangle is isosceles.
Area is :7
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

8. **Consider a publishing company that markets both book and audio cassette version to its works. Create a class Publication that stores the title (a string) and price (type float) of a publication. Derive the following two classes from the above Publication class: Book which adds a page count (int) and Tape which adds a playing time in minutes(float). Each class should have get_data() function to get its data from the user at the keyboard. Write the main() function to test the Book and Tape classes by creating instances of them asking the user to fill in data with get_data() and then displaying it using put_data().**

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

```
{ private:
string tital;
float price;
 public:     void
get_data_p()
    {
      cout<<"Enter the tital :";
cin>>tital;
      cout<<"Enter the price :";
cin>>price;
    }
    void put_data_p()
    {
      cout<<"Your tital is :"<<tital<<endl;
cout<<"Your price of book :"<<price<<endl;
    }   };
class book : public publication
{
  int count;
   public:     void
get_data_b()
    {
get_data_p();
      cout<<"Enter the page count :";
cin>>count;
    }
    void put_data_b()
    {
put_data_p();
      cout<<"Your page count is :"<<count <<end;
    } };
class tape : public publication
{   float
playtime;
public:
  void get_data_t()
  {     get_data_p();
cout<<"Enter the playtime :";
cin>>playtime;
  }
  void put_data_t()
  {
put_data_p();
    cout<<"Enter the playtime is :"<<playtime;
  }
```

```
}; int main() {   tape t;
book b;   cout<<"for
book:"<<endl;
  b.get_data_b();
  b.put_data_b();
cout<<"for tape:"<<endl;
  t.get_data_t();
  t.put_data_t();
return 0;
}
```

OUTPUT :-

```
for book:
Enter the tital :Bhagavad_Gita
Enter the price :1450
Enter the page count :1088
for tape:
Enter the tital :108
Enter the price :1450
Enter the playtime :30min
Your tital is :108
Your price of book :1450
Enter the playtime is :30
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

9. **Consider an example of declaring the examination result. Design three classes student, exam and result. The student has data members such as rollno, name. Create the lass exam by inheriting the student class. The exam class adds data members representing the marks scored in 5 subjects. Derive the result from exam-class and it has own data members like total, avg.**

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

class Student
{   protected:
int rollNo;
string name;
   public:     void setData(int
r, string n)
    {
rollNo = r;
name = n;
    }
};
```

```
class Exam : public Student
{
protected:
    int marks[5];
   public:     void
setMarks(int m[])
    {
        for (int i = 0; i < 5; i++)
        {
            marks[i] = m[i];
        }
    }
};

class Result : public Exam
{   private:
int total;
float avg;

public:
void
calculate
Result()
    {
total = 0;
        for (int i = 0; i < 5; i++)
        {
            total += marks[i];
        }
        avg = total / 5.0;
    }
    void displayResult()
    {
        cout << "Roll No: " << rollNo << endl;
cout << "Name: " << name << endl;         cout
<< "Marks: ";         for (int i = 0; i < 5;
i++)
        {
            cout << marks[i] << " ";
        }
cout << endl;
          cout << "Total Marks: " << total << endl;
      cout << "Average Marks: " << avg << endl;
    }
};
```

```
 int
main()
{
    Result studentResult;
    int marks[] = {85, 90, 75, 88, 92};

    studentResult.setData(101, "Vinod kumar");
studentResult.setMarks(marks);
studentResult.calculateResult();
studentResult.displayResult();

return 0;
}
```

OUTPUT :-

```
Roll No: 101
Name: Vinod kumar
Marks: 85 90 75 88 92
Total Marks: 430
Average Marks: 86
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

10. Write a program for overloading of Unary++ operator.

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

class distance1
{   public:     int feet,
inch;     distance1(int f,
int i)
    {
feet = f;
inch = i;
    }
    void operator-()
    {
feet--;
inch--;
        cout << "feet and inches: " << feet << " & " << inch;
    }
};
 int
main() {
```

```
    distance1 d1(10, 22);
    -d1;
return 0; }
```

OUTPUT :-

```
feet and inches: 9 & 21
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

11. Write a program for overloading of Binary + operator.

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

class MyDistance
{   public:     int
feet, inch;

public:
    MyDistance(int f=0, int i=0)
    {
feet = f;
inch = i;
    }

    MyDistance operator+(const MyDistance& d2)
    {
        MyDistance d3;
d3.feet = feet+d2.feet;
d3.inch = inch+d2.inch;
return d3;
    }
    void print()
    {
        cout <<"feet = "<<feet<<" & inch = "<<inch;
```

```
    }
};
 int
main()
{
    MyDistance d1(12, 8);
    MyDistance d2(10, 3);
    MyDistance d3 = d1+d2;

    cout<<"\nTotal Feet & Inches is:\n"<<d3.feet <<"  &  "
<<d3.inch<<endl;
d3.print();     return
0; }
```

OUTPUT :-

```
Total Feet & Inches is:
22  &  11
feet = 22 & inch = 11
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

12. Write a program of Virtual Functions.

```
#include<iostream>
using namespace std;
 class base {     public:
virtual void print()
      {
        cout<<"Print base class.";
      }
void show()
{
        cout<<"\nShow base class.";
      }
};

class Drived : public base
{     void
print()     {
        cout<<"Print drived class.";
    }     void
show()     {
        cout<<"\nPrint drived class.";
    } }; int
main() {    base
*obj;    Drived d;
obj = &d;    obj -
> print();    obj
-> show();
return 0;
```

```
}
```

OUTPUT :-

```
Print drived class.
Show base class.
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

13. Write a program of Abstract Classes.

```
#include<iostream>
using namespace std;
 class base {     public:
virtual void fun()=0;

};

class Derived : public base
{     public:
void fun()
{
        cout<<"Fun called.";
     } }; int
main() {
Derived d;
    d.fun();
return 0;
}
```

OUTPUT :-

```
Fun called.
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```

14. Write a program to read and write from file.

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
 int
main()
{
    // Open input file     ifstream
inputFile("input.txt");     // Check
if the input file is open     if
(!inputFile.is_open())
    {
        cerr << "Error: Unable to open input file." << endl;
return 1;
    }
    // Open output file     ofstream
outputFile("output.txt");     // Check
if the output file is open     if
(!outputFile.is_open())
    {
        cerr << "Error: Unable to open output file." << endl;
return 1;
    }     string
line;
    // Read data from input file line by line
while (getline(inputFile, line))
    {
        // Manipulate the data (e.g., convert to uppercase)
for (char &c : line)
        {
            c = toupper(c);
        }
        // Write the manipulated data to the output file
outputFile << line << endl;
    }
    // Close the input and output files
    inputFile.close();
outputFile.close();
```

```
    cout << "Data has been read from input.txt, manipulated,
and written to output.txt successfully." << endl;     return
0;
}
```

OUTPUT :-

```
Error: Unable to open input file.
PS C:\Users\gupta\OneDrive\Desktop\C++ Program>
```