

Basic and Advance C++ Programs

Function

- 11.** Define three functions named divide(). First function takes numerator and denominator as an input argument and checks if it's divisible or not, Second function takes one int number as input argument and checks whether the number is prime or not and Third function takes 3 float number as arguments, divide it with 3 and print average of the numbers Use concept of **Function Overloading / static binding**.

CODE:

```
#include<iostream>
int divide(int x,int y);
int divide(int x);
float divide(float x,float y,float z);
using namespace std;
int main()
{
    int x,y,p,q;
    float a,b,c,r;
    cout<<"enter value of a(numerator) & b(denominator) : \n PLEASE EXCEPT
DENOMINATOR AS \"0\" \n";
    cin>>x>>y;
    p=divide(x,y);
    if(p==1)
    {
        cout<<"the numbers are divisible";
    }
    else
    {
        cout<<"the numbers are not divisible";
    }
    cout<<"\n\nenter one num. for check num. is prime or not :";
    cin>>x;
    q=divide(x);
    if(q==(x-2))
    {
        cout<<"number is prime";
    }
    else
    {
        cout<<"number is not prime";
    }
}
```

Basic and Advance C++ Programs

```
    }
    cout<<"\n\nenter value of three numbers :\n";
    cin>>a>>b>>c;
    r=divide(a,b,c);
    cout<<"average : "<<r;
    cout<<"\n\nprepared by PRATIK DHORIYANI : 18CE024\n\n";
    return 0;
}
int divide(int x,int y)
{
    if(x%y==0)
    {
        return 1;
    }
    return 0;
}
int divide(int x)
{ int j=0;
  for(int i=2;i<x;i++)
  {
      if(x%i!=0)
      {
          j++;
      }
  }
  return j;
}
float divide(float x,float y,float z)
{
    float w;
    w=(x+y+z)/3;
    return w;
}
```

OUTPUT:

Basic and Advance C++ Programs

```
enter value of a(numerator) & b(denominator) :  
PLEASE EXCEPT DENOMINATOR AS "0"  
12  
6  
the numbers are divisible  
  
enter one num. for check num. is prime or not :7  
number is prime  
  
enter value of three numbers :  
2.2  
3.3  
4.4  
average : 3.3  
  
prepared by PRATIK DHORIYANI : 18CE024
```

12. Define four function void swap () which accepts two arguments by reference and swap the values. First function swaps two characters, second function swaps two integers, third function swaps two floats values and fourth function swaps two double values. Use the concept of **call by reference** in all four functions and **function overloading and inline function**.

CODE:

```
#include<iostream>  
inline void swap(char a,char b);  
inline void swap(int &p,int &q);  
inline void swap(float &c,float &d);  
inline void swap(double &r,double &s);  
using namespace std;  
int main()  
{  
    char x,y;  
    cout<<"enter two character x & y : \n";  
    char &a=x;  
    char &b=y;  
    cin>>x>>y;  
    swap(a,b);  
  
    int p,q;  
    cout<<"\n\nenter two integer p & q : \n";  
    cin>>p>>q;  
    swap(p,q);  
  
    float c,d;  
    cout<<"\n\nenter two float c & d : \n";
```

Basic and Advance C++ Programs

```
cin>>c>>d;
swap(c,d);

double r,s;
cout<<"\n\nenter two double value r & s : \n";
cin>>r>>s;
swap(r,s);

cout<<"\nprepared by PRATIK DHORIYANI : 18CE024\n\n";
return 0;

}
inline void swap(char a,char b)
{
    char temp;
    temp=a;
    a=b;
    b=temp;
    cout<<"Swaping of 2 character : \n"<<a<<"\n"<<b;

}
inline void swap(int &p,int &q)
{
    int temp;
    temp=p;
    p=q;
    q=temp;
    cout<<"Swaping of 2 integer : \n"<<p<<"\n"<<q;

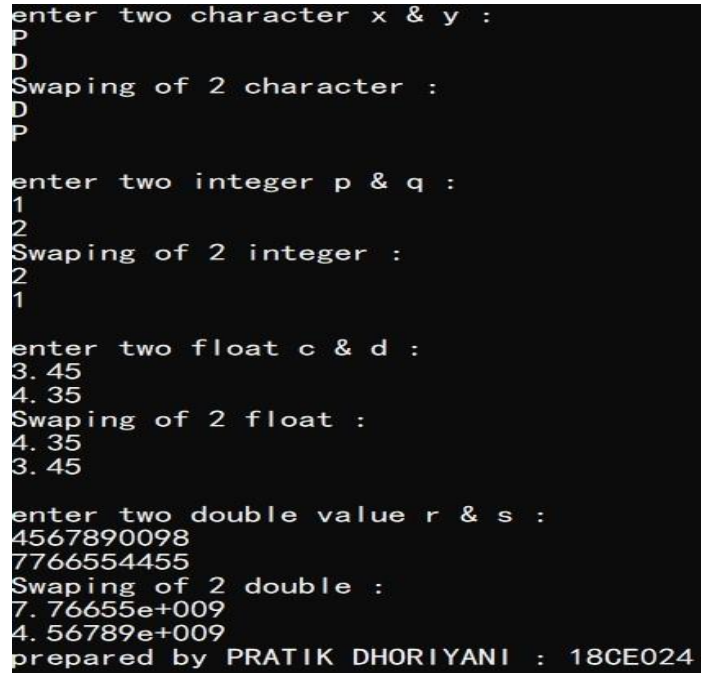
}
inline void swap(float &c,float &d)
{
    float temp;
    temp=c;
    c=d;
    d=temp;
    cout<<"Swaping of 2 float : \n"<<c<<"\n"<<d;

}
inline void swap(double &r,double &s)
{
    double temp;
```

Basic and Advance C++ Programs

```
temp=r;
r=s;
s=temp;
cout<<"Swaping of 2 double : \n"<<r<<"\n"<<s;
}
```

OUTPUT:



```
enter two character x & y :
P
D
Swaping of 2 character :
D
P

enter two integer p & q :
1
2
Swaping of 2 integer :
2
1

enter two float c & d :
3.45
4.35
Swaping of 2 float :
4.35
3.45

enter two double value r & s :
4567890098
7766554455
Swaping of 2 double :
7.76655e+009
4.56789e+009
prepared by PRATIK DHORIYANI : 18CE024
```

13. Write a function called `tonLarge()` that takes two integer arguments by **reference** and then sets the larger of the two numbers to 100 using **Return by reference**. Write a `main()` program to exercise this function.

CODE:

USING REFERENCE VARIABLE

```
#include<iostream>
int& tonlarge(int &x,int &y);
using namespace std;
int main()
{
    int a,b,p;
    cout<<"enter the values of a & b : \n";
    cin>>a>>b;
    tonlarge(a,b)=100;
    cout<<a<<"\n"<<b;
    cout<<"\nprepared by PRATIK DHORIYANI : 18CE024\n\n";
}
```

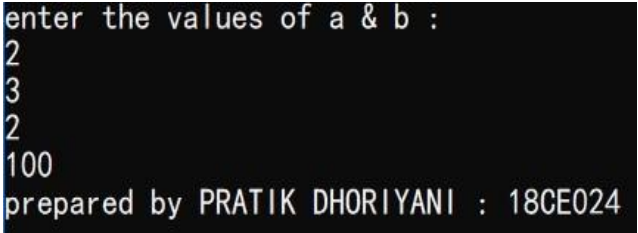
Basic and Advance C++ Programs

```
    return 0;
}
int& tonlarge(int &x,int &y)
{
    if(x>y)
        return x;
    else
        return y;
}
```

USING GLOBLE VARIABLE

```
#include<iostream>
int& tonlarge();
int a,b;
using namespace std;
int main()
{
    cout<<"enter the values of a & b : \n";
    cin>>a>>b;
    tonlarge();
    cout<<a<<"\n"<<b;
    cout<<"\nprepared by PRATIK DHORIYANI : 18CE024\n\n";
    return 0;
}
int& tonlarge()
{
    if(a>b)
        return a=100;
    else
        return b=100;
}
```

OUTPUT:

A screenshot of a terminal window showing the output of the C++ program. The text is as follows:

```
enter the values of a & b :
2
3
2
100
prepared by PRATIK DHORIYANI : 18CE024
```

Basic and Advance C++ Programs

- 14.** Write a function called `power ()` that takes two arguments: a double value for **n** and an int for **p**, and returns the result as double value. Use **default argument** of 2 for **p**, so that if this argument is omitted, the number will be squared. Write a `main ()` function that gets values from the user to test this function.

CODE:

```
#include<iostream>
#include<math.h>
double power(double n,int p=2);
using namespace std;
int main()
{
    int n,p,ans;
    cout<<"enter the values of n & p : \n";
    cin>>n>>p;
    ans=power(n);
    cout<<ans;
    cout<<"\nprepared by PRATIK DHORIYANI : 18CE024\n\n";
    return 0;
}
double power(double n,int p)
{
    double x=pow(n,p);
    return x;
}
```

OUTPUT:

```
enter the values of n & p :
3
2
9
prepared by PRATIK DHORIYANI : 18CE024
```

Classes and Objects

- 15.** Define a C++ **Structure** Rectangle with data member's width and height. It has `get_values()` member functions to get the data from user and `area()` member functions to print the area of rectangle. Also create a C++ **Class** for the above program. **Define both functions inside the class. Member function defined inside the class behaves like an inline function** and illustrate the difference between C++ Structure and C++ Class.

Basic and Advance C++ Programs

CODE:

USING STRUCTURE

```
#include<iostream>
using namespace std;
struct rectangle
{
    float width;
    float hight;
    void get_values()
    {
        cout<<"enter the values of WIDTH & HIGHT : \n";
        cin>>width>>hight;
    }
    void area()
    {
        cout<<"\nthe AREA of rectangle is : \n";
        cout<<width*hight;
    }
}r;
int main()
{
    r.get_values();
    r.area();
    cout<<"\nprepared by PRATIK DHORIYANI : 18CE024\n\n";
    return 0;
}
```

USING CLASS

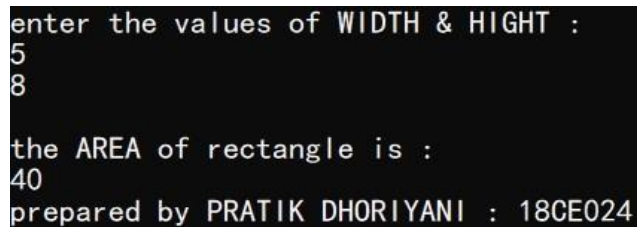
```
#include<iostream>
using namespace std;
class rectangle
{
    private:
        float width;
        float hight;

    public:
        void get_values()
        {
            cout<<"enter the values of WIDTH & HIGHT : \n";
```


Basic and Advance C++ Programs

```
        cin>>width>>hight;
    }
    void area()
    {
        cout<<"\nthe AREA of rectangle is : \n";
        cout<<width*hight;
    }
}r;
int main()
{
    r.get_values();
    r.area();
    cout<<"\nprepared by PRATIK DHORIYANI : 18CE024\n\n";
    return 0;
}
```

OUTPUT:

A screenshot of a terminal window showing the output of a C++ program. The text is as follows:
enter the values of WIDTH & HIGHT :
5
8

the AREA of rectangle is :
40

prepared by PRATIK DHORIYANI : 18CE024

Difference between C++ CLASS & STRUCTURE :

A) Class

1. A class in C++ can be defined as a collection of related variables and functions encapsulated in a single structure.
2. A class in C++ is just an extension of a structure used in the C language. It is a user defined data type. It actually binds the data and its related functions in one unit.
3. Keyword for the declaration: **Class**
4. Default access specifier: **Private**
5. Purpose: **Data abstraction and further inheritance**
6. Type: **Reference**
7. Usage: **Generally used for large amounts of data.**
8. Its object is created on the **heap memory**.
9. The member variable of class **can be initialized directly**.
10. It can have all the types of **constructor and destructor**.

B) Structure

1. A structure in C++ can be referred to as an user defined data type possessing its own operations.

Basic and Advance C++ Programs

2. A structure and a class in C language differs a lot as a structure has limited functionality and features as compared to a class.
3. Keyword for the declaration: **Struct**
4. Default access specifier: **Public**
5. Purpose: **Generally, grouping of data**
6. Type: **Value**
7. Usage: **Generally used for smaller amounts of data.**
8. Its object is created on the **stack memory**.
9. The member variable of structure **cannot be initialized directly**.
10. It can have only **parameterized constructor**.

16. Write a C++ program having class **Batsman**. It has private data members: batsman name, bcode (4 Digit Code Number), innings, not out, runs, batting average. Innings, not out and runs are in integer and batting average is in float.

Define following **function outside the class using scope resolution operator**.

- 1) Public member function getdata() to read values of data members.
- 2) Public member function putdata() to display values of data members.
- 3) **Private member function** calcavg() which calculates the batting average of a batsman. Also make this outside function **inline**. *Hint : batting average = runs/(innings - notout)*

CODE:

```
#include<iostream>
using namespace std;
class batsman
{
    private:
        char batsname[15];
        int bcode;
        int runs;
        float battingavg;
        int innings;
        int notout;

        inline void calcavg();
    public:
        void getdata();
        void putdata();
}b;
void batsman::getdata()
{
```

Basic and Advance C++ Programs

```
        cout<<"enter the value for BATSMAN NAME , BATSMAN CODE , RUNS , INNINGS ,  
NOTOUT : \n";  
        cin>>batsname>>bcode>>runs>>innings>>notout;  
    }  
void batsman::putdata()  
{  
    cout<<"OUTPUT : \n";  
    cout<<batsname<<"\n"<<bcode<<"\n"<<runs<<"\n"<<innings<<"\n"<<notout<<"\n";  
    calcavg();  
}  
inline void batsman::calcavg()  
{  
    battingavg=runs/(innings - notout);  
    cout<<"BATTING AVERAGE : "<<battingavg;  
}  
int main()  
{  
    b.getdata();  
    b.putdata();  
    cout<<"\n\nprepared by PRATIK DHORIYANI : 18CE024\n\n";  
    return 0;  
}
```

OUTPUT:

```
enter the value for  BATSMAN NAME , BATSMAN CODE , RUNS , INNINGS , NOTOUT :  
pratik  
1234  
1000  
16  
6  
OUTPUT :  
pratik  
1234  
1000  
16  
6  
BATTING AVERAGE : 100  
prepared by PRATIK DHORIYANI : 18CE024
```

17. Define class Digit having 'int n' as data member. Define member function enter() to enter the data and show() to print the data. A class has member function compare() that displays whether the first object is smaller, greater or same as compared to second object. (Function compare() should support: int x = d1.compare(d2); where d1 and d2 are objects of class Digit). Use Concept of Object as Function Arguments.

CODE:

Basic and Advance C++ Programs

```
#include<iostream>
using namespace std;
class digit
{
    int n;
public:
    void enter()
    {
        cin>>n;
    }
    void show()
    {
        cout<<n;
    }
    int compare(digit);
};
int digit::compare(digit d)
{
    if(n==d.n)
    {
        return 0;
    }
    else if(n>d.n)
    {
        return 1;
    }
    else if(n<d.n)
    {
        return -1;
    }
}
int main()
{
    digit d1,d2;
    cout<<"enter value of n for D1 : ";
    d1.enter();
    d1.show();
    cout<<"\nenter value of n for D2 : ";
    d2.enter();
    d2.show();
}
```

Basic and Advance C++ Programs

```
int x = d1.compare(d2);
if(x==0)
    cout<<"\nboth D1 & D2 are EQUAL";
else if(x==1)
    cout<<"\nD1 is GRATER than D2";
else if(x==-1)
    cout<<"\nD1 is LESS than D2";
cout<<"\n\nprepared by PRATIK DHORIYANI : 18CE024\n\n";
return 0;
}
```

OUTPUT:

```
enter value of n for D1 : 8
8
enter value of n for D2 : 9
9
D1 is LESS than D2
prepared by PRATIK DHORIYANI : 18CE024
```

18. Define class *Currency* having two integer data members rupee and paisa. A class has member functions *enter()* to get the data and *show()* to print the amount in 22.50 format. Define one member function *sum()* that adds two objects of the class and stores answer in the third object i.e. *c3=c1.sum (c2)*. The second member function should add two objects of type *currency* passed as arguments such that it supports *c3.sum (c1,c2)*; where *c1*, *c2* and *c3* are objects of class *Currency*. Also Validate your answer if paisa >100. Write a *main()* program to test all the functions. **Use concepts of Object as Function Arguments, function returning object and function overloading.**

CODE:

```
#include<iostream>
#include<iomanip>
using namespace std;
class currency
{
    int rupee,paisa;
public:
    void enter()
    {
        cin>>rupee>>paisa;
    }
    void show()
```

Basic and Advance C++ Programs

```
{
    float amount=float((rupee*100) + paisa)/100;
    cout<<"\nthe sum of c1 & c2 (TOTAL AMOUNT) is :\n";
    cout<<fixed<<setprecision(2)<<amount;
}
currency sum(currency);
void sum(currency,currency);
};
currency currency::sum(currency p2)
{
    currency temp;
    temp.rupee=rupee + p2.rupee;
    temp.paisa=paisa + p2.paisa;
    return temp;
}
void currency::sum(currency d1,currency d2)
{
    rupee=d1.rupee + d2.rupee;
    paisa=d1.paisa + d2.paisa;
}
int main()
{
    currency c1,c2,c3;
    cout<<"enter rupee and paisa for c1 : \n";
    c1.enter();
    cout<<"enter rupee and paisa for c2 : \n";
    c2.enter();
    c3=c1.sum(c2);
    c3.show();
    c3.sum(c1,c2);
    c3.show();
    cout<<"\n\nprepared by PRATIK DHORIYANI : 18CE024\n";
    return 0;
}
```

OUTPUT:

Basic and Advance C++ Programs

```
enter rupee and paisa for c1 :
20
60
enter rupee and paisa for c2 :
30
70

the sum of c1 & c2 (TOTAL AMOUNT) is :
51.30
the sum of c1 & c2 (TOTAL AMOUNT) is :
51.30

prepared by PRATIK DHORIYANI : 18CE024
```

19. Define a class **Dist** with int feet and float inches. Define member function that displays distance in 1"-2.5" format. Also define member function scale () function that takes **object by reference** and scale factor in float as an input argument. The function will scale the distance accordingly.

For example, 20"-5.5" and Scale Factor is 0.5 then answer is 10"-2.75"

CODE:

```
#include<iostream>
using namespace std;
class dist
{
    int feet;
    float inches;
public:
    void get()
    {
        cout<<"\nenter FEET & INCHES : \n";
        cin>>feet>>inches;
    }
    void scale(dist &, float);
    void print()
    {
        cout<<"the final DISTANCE is : ";
        cout<<"\n"<<feet<<"-"<<inches<<"\n";
    }
};

void dist::scale(dist &p, float s)
{
    feet=p.feet*s;
    inches=p.inches*s;
}
```

Basic and Advance C++ Programs

```
int main()
{
    dist d;
    float sf;
    cout<<"enter value for SCALE FACTOR : ";
    cin>>sf;
    d.get();
    d.scale(d,sf);
    d.print();
    cout<<"\n\nprepared by PRATIK DHORIYANI : 18CE024\n";
    return 0;
}
```

OUTPUT:

```
enter value for SCALE FACTOR : 0.5
enter FEET & INCHES :
10
5.5
the final DISTANCE is :
5'-2.75"
prepared by PRATIK DHORIYANI : 18CE024
```

- 20.** Create a Class Gate for students appearing in Gate (Graduate Aptitude test for Engineering) exam. There are three examination center Vadodara, Surat, and Ahmedabad where Gate exams are conducted. A class has data members: Registration number, Name of student, Examination center. Class also Contains static data member ECV_Cnt, ECS_Cnt and ECA_Cnt which counts the number of students in Vadodara, Surat and Ahmedabad exam center respectively. Class Contains two Member function getdata () which gets all information of students and counts total students in each exam center and pudata () which prints all information about the students. Class also contains one static member function getcount () which displays the total number of students in each examination center. Write a program for 5 students and display the total number of students in each examination center. **Use static data member, static member function and Array of Objects.**

CODE:

```
#include<iostream>
#include<string.h>
using namespace std;
struct gate
{
    int regno;
```


Basic and Advance C++ Programs

```
char name[10];
char excen[10];
static int ECV_Cnt;
static int ECS_Cnt;
static int ECA_Cnt;
public:
void getdata()
{
    cin>>regno>>name>>excen;

    if(strcmp(excen,"vadodara")==0)
    {
        ECV_Cnt++;
    }
    else if(strcmp(excen,"surat")==0)
    {
        ECS_Cnt++;
    }
    else if(strcmp(excen,"ahmedabad")==0)
    {
        ECA_Cnt++;
    }
}
static void getcount()
{
    cout<<"total students in vadodara : "<<ECV_Cnt<<"\n";
    cout<<"total students in surat : "<<ECS_Cnt<<"\n";
    cout<<"total students in ahmedabad : "<<ECA_Cnt<<"\n\n";
}
void putdata()
{
    cout<<"REGISTRATION NO. : "<<regno<<"\n"<<"STUDENT NAME : "<<name<<"\n";
}
};
int gate::ECV_Cnt;
int gate::ECS_Cnt;
int gate::ECA_Cnt;

int main()
{
```

Basic and Advance C++ Programs

```
gate o[5];
cout<<"enter input for REGISTRATION NO. , NAME , CENTER NAME of 5 students : \n";
for(int i=0;i<5;i++)
{
    o[i].getdata();
}
cout<<"\n";
for(int i=0;i<5;i++)
{
    o[i].putdata();
    cout<<"\n";
}
cout<<"\n";
gate::getcount();
cout<<"\nprepared by PRATIK DHORIYANI : 18CE024\n\n";
return 0;
}
```

OUTPUT:

```
enter input for REGISTRATION NO. , NAME , CENTER NAME of 5 students :
1
pratik
surat
2
dhruv
surat
3
rudra
surat
4
kelvin
surat
5
vratik
vadodara

REGISTRATION NO. : 1
STUDENT NAME : pratik

REGISTRATION NO. : 2
STUDENT NAME : dhruv

REGISTRATION NO. : 3
STUDENT NAME : rudra

REGISTRATION NO. : 4
STUDENT NAME : kelvin

REGISTRATION NO. : 5
STUDENT NAME : vratik

total students in vadodara : 1
total students in surat : 4
total students in ahmedabad : 0

prepared by PRATIK DHORIYANI : 18CE024
```