UNIT : 1

1) Write a C++ program to find a number is prime or not

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int n;
    cout<<"Enter a postitve integer  ";
    cin>>n;
    if(n<2)
    {
        cout<<"Invalid Entry........";
        getch();
        return -1;
    }
    int flag=0;
    for(int i=2;i<n;i++)
    {
        if(n%i==0)
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
    cout<<n<<"  Not a Prime Number";
    else
    cout<<n<<"  Is a Prime Number";
    getch();
    return 0;
}
```

2)Write a C++ program to find the reverse of a number. Hence find it is palindrome or not?

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int n;
    cout<<"Enter a number : ";
    cin>>n;
    int d,rev=0,num=n;
    do
    {
        d=n%10;
        rev=rev*10+d;
        n=n/10;
    }
    while(n>0);
    cout<<num<<"\n";
    cout<<rev<<"\n";

    if(num==rev)
    cout<<"It is a Palindrome";
    else
    cout<<"It is not a Palindrome";
    return 0;
}
```

3) Write a C++ program to find square of a number

```
#include<iostream>
#include<math.h>

using namespace std;

int main()
{
   float sq,n;
   cout<<"Enter any number:";
   cin>>n;
   sq=sqrt(n);
   cout<<"Square root of "<<n<<" is "<<sq;
   return 0;
}
```

4) Write a C++ program to add two numbers using a method that takes two numbers as parameters and returns the result

```
#include <iostream>
#include <conio.h>
using namespace std;
int add(int a,int b)
{
    return a+b;
}
int main()
{
    cout<<"Enter two numbers : ";
    int x,y;
    cin>>x>>y;
    int z=add(x,y);
    cout<<x<<"+"<<y<<"="<<z;
    getch();
    return 0;
}
```

5) Write a C++ program to find the factorial of a number using recursive function

```
#include <iostream>
#include <conio.h>
using namespace std;
long fact(int n)
{
    if(n==0)
    return 1;
    else
    return n*fact(n-1);
}
int main()
{
    int x;
    cout<<"Enter x : ";
    cin>>x;
    if(x<0)
    cout<<"Factorial is not defined for -ve number";
    else
    cout<<x<<"!="<<fact(x);
    getch();
    return 0;
}
```

6) Write a C++ program to swap two numbers using call by reference

```
#include <iostream>
#include <conio.h>
using namespace std;
void swap(int &a,int &b)
{
    int c=a;
    a=b;
    b=c;
}
int main()
{
    int x,y;
    cout<<"Enter 1st number : ";
    cin>>x;
    cout<<"Enter 2nd number : ";
    cin>>y;
    cout<<"Before Swapping \n";
    cout<<"x="<<x<<"\n";
    cout<<"y="<<y<<"\n";
    swap(x,y);
    cout<<"After Swapping "<<"\n";
    cout<<"x="<<x<<"\n";
    cout<<"y="<<y<<"\n";
}
```

UNIT : 2

1)Write a C++ program to design an Employee class for reading and displaying the employee information (employee ID and employee name), with the getInfo() and displayInfo() methods respectively. Where getInfo() will be a private method.

```
#include <iostream>
using namespace std;
class Employee
{
  int empid;
  char empname[20];
  void getInfo()
  {
      cout<<"Employee Id  ";
      cin>>empid;
      cout<<"Employee Name  ";
      cin>>empname;
  }
  public:
  Employee ()
  {
      getInfo();
  }
  void displayInfo()
  {
     cout<<empid<<"\t"<<empname<<"\n";
  }
};
int main()
{
    cout<<"Enter 1st employee details\n";
    Employee e1;
    cout<<"Enter 2nd employee details\n";
    Employee e2;
    cout<<"Displaying Details\n";
    e1.displayInfo();
    e2.displayInfo();
    return 0;
}
```

2) Write a C++ program to design the class Student containing data members Rollno, name and age. Include a method named display() for display the student information. Include default constructor and parameterized constructor with default argument to initialize the data members.
Display three students' details, using array of objects

```
#include<iostream>
#include<string.h>
using namespace std;
class Student
{
    int rollno,age;
    char name[20];
    public:
    Student()
    {
        cout<<"Enter rollno name and age  ";
        cin>>rollno>>name>>age;
    }
    Student(int r,char *n,int a=18)
    {
        rollno=r;
        strcpy(name,n);
        age=a;
    }
    void display()
    {
        cout<<"\n\t"<<rollno<<"\t"<<name<<"\t"<<age;
    }
};
int main()
{
    Student st[3]={{},{102,"xyz"},{103,"pqr",20}};
    cout<<"Details are\n";
    cout<<"\n\trollno\tname\tage\n";
    for(int i=0;i<3;i++)
    st[i].display();
    return 0;
}
```

3)Write a C++ Program to demonstrate function definition outside class and accessing class members in function definition

- Create a class named Rectangle with two data members' length and breadth
- Include a constructor to initialize the data members and a method to find the area of rectangle. Define both the constructor and member function outside the class

```
#include<iostream>
#include<string>
using namespace std;
class Rectangle
{
    int length,breadth;
    public:
    Rectangle(int,int);
    int findarea();
};
Rectangle::Rectangle(int length,int breadth)
{
    this->length=length;
    this->breadth=breadth;
}
int Rectangle::findarea()
{
    return length*breadth;
}
int main()
{
    cout<<"To find area of rectangle\n";
    int l,b;
    cout<<"Enter length:";
    cin>>l;
    cout<<"Enter breadth:";
    cin>>b;
    Rectangle r1(l,b);
    cout<<r1.findarea();
    return 0;
}
```

4)Write a C++ program to design a class Geometry containing the methods

- area() for finding the area of a square
- volume() for finding the volume of a cuboid.

Also overload the area() function for finding the area of a rectangle

```
#include <iostream>
#include <conio.h>
using namespace std;
class Geometry
{
    public:
    int area(int x)
    {
        return x*x;
    }
    int volume(int x,int y,int z)
    {
        return x*y*z;
    }
    int area(int x,int y)
    {
        return x*y;
    }
};
int main()
{
    Geometry obj;
    int side;
    cout<<"Enter side : ";
    cin>>side;
    cout<<"Area of square : "<<obj.area(side);
    int l,b,h;
    cout<<"\nVolume of cube..... \nEnter length ";
    cin>>l;
    cout<<"Enter breadth";
    cin>>b;
    cout<<"Enter height";
    cin>>h;
    cout<<"Volume of cube : "<<obj.volume(l,b,h);
    cout<<"\nArea of rectangle.....\n Enter length";
    cin>>l;
    cout<<"Enter breadth";
    cin>>b;
    cout<<"Area of rectangle :"<<obj.area(l,b);
    return 0;
}
```

5)Write C++ program to design a class Rectangle to show the implementation of static variable and static function; that can be used to count the number of rectangle object created.

```
#include <iostream>
#include <conio.h>
using namespace std;
class Rectangle
{
  int length,breadth;
  static int count;
  public:
  Rectangle(int,int);
  void findarea();
  static int getcount();
};
Rectangle::Rectangle(int length,int breadth)
{
    this->length=length;
    this->breadth=breadth;
    count++;
}
void Rectangle::findarea()
{
    cout<<"\n\tLength  : "<<length;
    cout<<"\n\tBreadth : "<<breadth;
    cout<<"\n\tArea    : "<<length*breadth;
}
int Rectangle::getcount()
{
    return count;
}
int Rectangle::count=0;
int main()
{
 cout<<"\nNo. of rectangle object is : "<<Rectangle::getcount();
 Rectangle r1(10,2),r2(4,3);
 cout<<"\nRectangle 1";
 r1.findarea();
 cout<<"\nRectangle 2";
 r2.findarea();
 cout<<"\nNo. of rectangle object is : "<<Rectangle::getcount();
 return 0;
}
```

1) Write a C++ program to overload unary minus (-) operator to negate a point in three dimensional space using a member function

```
#include <iostream>
#include <conio.h>
using namespace std;
class space
{
    int x,y,z;
    public:
    space(int x,int y,int z)
    {
       this -> x=x;
       this -> y=y;
       this -> z=z;
    }
    void display()
    {
        cout<<x<<",";
        cout<<y<<",";
        cout<<z<<",";
    }
    void operator -()
    {
        x=-x;
        y=-y;
        z=-z;
    }
};
int main()
{
    space s(10,-10,20);
    s.display();
    cout<<"\n";
    -s;
    s.display();
    getch();
    return 0;
}
```

2) Write a C++ program to overload the operator + for adding the timings of two clock objects using a member function

```
#include <iostream>
#include <conio.h>
using namespace std;
class Clock
{
    int hr,min,sec;
    public:
    Clock(){   }
    Clock(int hr,int min,int sec)
    {
       this -> hr=hr;
       this -> min=min;
       this -> sec=sec;
    }
    Clock display()
    {
        cout<<hr<<",";
        cout<<min<<",";
        cout<<sec<<",";
    }
    Clock operator +(Clock c)
    {
       int a,b,h;
       a=sec+c.sec;
       b=min+c.min+(a/60);
       h=hr+c.hr+(b/60);

       int s=a%60;
       int m=b%60;

       Clock tmp(h,m,s);
       return tmp;

    }
};
int main()
{
   Clock c1(16,15,25),c2(8,45,55),
    c3=c1+c2;
    c1.display();
     cout<<"\n";
    c2.display();
     cout<<"\n";
    c3.display();
    getch();
    return 0;
}
```

3)Write a C++ program to overload the + operator for concatenating two strings using member function. For e.g "I Love " + "India" = I Love India

```
#include <iostream>
#include <conio.h>
#include <string.h>
using namespace std;
class String
{
    char *str;
    public:
    String(){   }
    String(char *s)
    {
        int len=strlen(s);
        str=new char[len+1];
        strcpy(str,s);
    }
    void display()
    {
        cout<<str;
    }
    String operator +(String s)
    {
        String tmp;
        int len=strlen(str) + strlen(s.str);

        tmp.str=new char[len+1];
        strcpy(tmp.str,str);
        strcat(tmp.str,s.str);
        return tmp;
    }
};
int main()
{
    String s1("I am "),s2("Gaurang"),
    s3=s1+s2;
    s1.display();
     cout<<"\n";
    s2.display();
     cout<<"\n";
    s3.display();
    getch();
    return 0;
}
```

4) Implement the concept of method overriding in C++
Design a class named Employee with data members empcode, name & age and methods get and show to accept and display employee details
Design a class named ContractEmployee(with data member contractid) derived from Employee class and override methods get and show
Using ContractEmployee object, accept and display all details of ContractEmployee

```
#include <iostream>
#include <string.h>

using namespace std;
class Employee
{
    int empcode,age;
    string name;
    public:
    void get()
    {
        cout<<"Employee Code";
        cin>>empcode;
        cout<<"Name";
        cin>>name;
        cout<<"Age";
        cin>>age;
    }
  void show()
  {
      cout<<"\n"<<empcode;
      cout<<"\n"<<name;
      cout<<"\n"<<age;
  }
};
class ContractEmployee:public Employee
{
    int contactid;
    public:
    void get()
    {
        Employee::get();
        cout<<"Enter contact ID:";
        cin>>contactid;
    }
    void show()
    {
        Employee::show();
        cout<<"\n"<<contactid;
    }
};

int main()
{
    ContractEmployee obj;
    cout<<"\n Enter Employee Details:...\n";
    obj.get();
    cout<<"\nEmployee Details:...\n";
    obj.show();
    return 0;
}
```

5) Implement the concept of virtual function. Design three classes named Person (with data member name), Student (with data member CGPA) and Professor (with data member appraisal_score). Both Student and Professor must derive from Person class. Include a virtual function named getData() to accept values for class data members and a pure virtual function named findOutstanding() to return performance status of Student and Professor depending on CGPA and appraisal_score value.

```
#include<iostream>
#include<string.h>
using namespace std;
class Person
{
  char name[20];
  public:
  virtual void getData()
  {
      cout<<"Name ";
      cin>>name;
  }
  virtual string findoutstanding()=0;
};
class student:public Person
{
    float CGPA;
    public:
    void getData()
    {
        Person::getData();
        cout<<"CGPA ";
        cin>>CGPA;
    }
    string findoutstanding()
    {
        if(CGPA>=7.5) return "Outstanding\n";
        else return "Not Outstanding\n";
    }
};
class Professor:public Person
{
    float apprisalscore;
    public:
    void getData()
    {
        Person::getData();
        cout<<"apprisalscore ";
        cin>>apprisalscore;
    }
    string findoutstanding()
    {
        if(apprisalscore>=75) return "Outstanding\n";
        else return "Not Outstanding\n";
    }
};
int main()
{
    Person *obj;
    student st;
    Professor pf;
    cout<<"Enter student details\n";
    obj=&st;
    obj->getData();
    cout<<obj->findoutstanding();
    cout<<"Enter Professor details\n";
    obj=&pf;
    obj->getData();
    cout<<obj->findoutstanding();
    return 0;
}
```

6) Implement the concept of abstract class. Design an abstract class named Shape(with data member area) and two pure virtual function named findarea() and displaydetails(). Create two classes named Rectangle (with data members length and breadth) and Circle (with data member radius) derived from Shape. Include constructor in both subclasses to initialize respective data members. Using the object of Rectangle and Circle, find and display the area of rectangle and circle respectively

```
#include<iostream>
using namespace std;
class Shape
{
  public:
  float area;
  virtual void findarea()=0;
  virtual void displaydetails()=0;
};
class Rectangle:public Shape
{
  float length,breadth;
  public:
  Rectangle(float l, float b)
  {
      length=l;
      breadth=b;
  }
  void findarea()
  {
      area=length*breadth;
  }
  void displaydetails()
  {
      cout<<"\tLength   :  "<<length<<"\n";
      cout<<"\tBreadth  :  "<<breadth<<"\n";
      cout<<"\tArea     :  "<<area<<"\n";
  }
};
class circle:public Shape
{
    float r;
    public:
    circle(float r)
    {
        this->r=r;
    }
    void findarea()
    {
        area=3.14*r*r;
    }
    void displaydetails()
    {
        cout<<"\tRadius  :  "<<r<<"\n";
        cout<<"\tArea    :  "<<area<<"\n";
    }
};
int main()
{
    Shape *obj;
    Rectangle r1(5.3,4.5);
    obj=&r1;
    cout<<"Rectangle.....\n";
    obj->findarea();
    obj->displaydetails();
    circle c1(2.5);
    obj=&c1;
    cout<<"Circle....\n";
    obj->findarea();
    obj->displaydetails();
    return 0;
}
```

1) Write a C++ program to find area of circle using single inheritance such that base class function accept radius from user and derived class function calculate and display area, using public type derivation

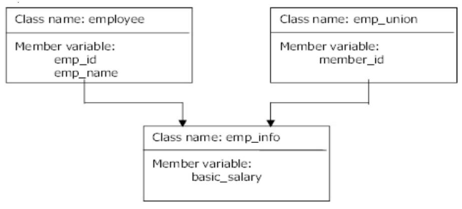
```
#include <iostream>
using namespace std;
class Circle1
{
    int radius;
    public:
    void setRadius()
    {
        cout<<"Enter Radius";
        cin>>radius;
    }
    int getRadius()
    {
        return radius;
    }
};
class Circle2:public Circle1
{
        public:
        void findarea()
        {
            int r=getRadius();
            cout<<"Radius : "<<r;
            float area=3.14*r*r;
            cout<<"\nArea : "<<area;
        }

};

int main()
{
    Circle2 obj;
    obj.setRadius();
    obj.findarea();
    return 0;

}
```

2) Write a C++ program to implement multiple inheritance as shown in figure. Include constructor to initialize data members and method to display detailsUsing array of emp_info objects, display details of two employees



```
#include <iostream>
#include<string.h>
using namespace std;
class Employee
{
  int emp_Id;
  char emp_Name[10];
  public:
  Employee(int id,char *name)
  {
      emp_Id=id;
      strcpy(emp_Name,name);
  }
  void dispEmployee()
  {
      cout<<"\n"<<emp_Id;
      cout<<"\n"<<emp_Name;
  }
};
class Emp_Union
{
    char member_Id[10];
    public:
    Emp_Union(char *mid)
    {
        strcpy(member_Id,mid);
    }
    void dispunion()
    {
        cout<<"\n"<<member_Id;
    }
};
class Emp_Info:public Employee,public Emp_Union
{
  int basic_Salary;
  public:
  Emp_Info(int id,char *name,char *mid,int salary):
  Employee(id, name), Emp_Union(mid)
  {
      basic_Salary=salary;
  }
  void display()
  {
      dispEmployee();
      dispunion();
      cout<<"\n"<<basic_Salary;
  }
};
int main()
{
        Emp_Info obj[2]={ {001, "Gaurang", "Mumbai", 20000},{002, "Hiren", "CurryRoad",
25000}};
   obj[0].display();
   obj[1].display();
   return 0;
}
```

3)Implement hierarchical inheritance
Write a C++ program to design a class "Number" with a private data member to read a number from user. Design a class "Square" inherited from "Number" to calculate square of number. Design another class "Cube" inherited from "Number" to calculate cube of number.

```
#include <iostream>
 using namespace std;
 class Number
 {
     int n;
     protected:
     int getN()
     {
         return n;
     }
     public:
     Number()
     {
         cout<<"\nEnter number:";
         cin>>n;
     }

 };
 class Square:public Number
 {
     public:
     void findsquare()
     {
         int n=getN();
         cout<<n*n;
     }

 };
 class Cube:public Number
 {
     public:
     void findcube()
     {
         int n=getN();
         cout<<n*n*n<<"\n";
     }
 };

 int main()
 {
    Square s1;
    s1.findsquare();
    Cube c1;
    c1.findcube();
     return 0;
 }
```

4)Write a program in C++ to obtain two numbers from user. If numbers are equal, display appropriate message else throw and handle an exception

```
#include<iostream>
using namespace std;
int main()
{
    int a,b;
    cout<<"Enter two numbers";
    cin>>a>>b;
    try
    {
        if(a==b)
        cout<<"Both are equal";
        else if(a>b) throw a;
        else throw b;
    }
    catch(int x)
    {
        cout<<"Number not equal\n";
        cout<<"Greater number"<<x;
    }
    return 0;
}
```

5) Write a program in C++ to accept the course name from a student. If course name is either “BScIT”, “BMS” or “BCom” display message with selected course, else throw and handle an exception
or
Write a program in C++ to country name from a user. If country name is “India” display welcome message, else throw and handle an exception

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string s;
    cout<<"Enter course name";
    cin>>s;
    try
    {
        if(s=="BScIT" || s=="BMS" || s=="BCom")
        cout<<"You have selected course:"<<s<<endl;
        else throw s;
    }
    catch(string str)
    {
        cout<<"Wrong Entry\n"<<str<<endl;
    }
    return 0;
}
```

1) Write a C++ program to count total number of characters, words and lines in a file

2) Write a C++ program to read the content of files “country”, “capital”, and “language” and display output as “Capital of Maharashtra is Mumbai where Marathi is spoken” on console. Display 5 such details

3) Write a C++ program to insert student details (rollno, name, percentage) into a file “stu.txt. Also display all students details from the file

5) Write a C++ program to create a function template for finding smallest element from an array of generic type

OR

Write a C++ program to create a function template for finding largest element from an array of generic type

6) Write a C++ program to design a class template for sorting in ascending order of generic type

4) Write a C++ program to design a function template for swapping numbers of generic type

```
#include<iostream>
using namespace std;
template<class type1>
void do_swap(type1 &a,type1 &b)
{
    type1 t=a;
    a=b;
    b=t;

}
int main()
{
    int x1,y1;
    cout<<"Enter two integers..  \n";
    cin>>x1>>y1;
    cout<<"\tIntegers before swapping\n";
    cout<<"\tx1="<<x1<<" and y1="<<y1;
    do_swap<int>(x1,y1);
    cout<<"\n\tIntegers after swapping\n";
    cout<<"\tx1="<<x1<<" and y1="<<y1;

    float x2,y2;
    cout<<"\nEnter two float numbers..\n";
    cin>>x2>>y2;
    cout<<"\tfloat before swapping\n";
    cout<<"\tx2="<<x2<<" and y2="<<y2;
    do_swap<float>(x2,y2);
    cout<<"\n\tfloat after swapping\n";
    cout<<"\tx2="<<x2<<" and y2="<<y2;
    return 0;

}
```