Assignment-2

OOPs Concept

Shubham Garg
9919103057
Batch: F2

1.

Procedural Language	Object Oriented Language
* Here, program is divided into small parts called functions.	* Here, program is divided into small parts called objects.
* Follows top down approach	* Follows bottom up approach
* No access specifiers in this programming.	* This have access specifiers like public, private, protected.
* Adding new data and functions is not Easy.	* Adding new data and functions is Easy.
* We can't hide anything. So, it is less secure.	* It provides data hiding. So, it is more secure.
* Overloading is not possible	* Overloading is possible
Ex - C, FORTRAN, Pascal etc.	Ex - C++, Python, Java, C# etc

2. C++ is object oriented and it is related to real world objects while C in procedural oriented so, it focus on procedure.

* C++ uses Inheritance while C doesnot.
* Overloading is allowed in C++
* C++ is enriched with access specifiers - public, private, protected.
* It is useful for low level programming language and very efficient for general purpose.

Shubham 9919103057

3. The type of Inheritance supported in C++ are –
* Single Inheritance
* Multilevel Inheritance
* Multiple Inheritance
* Heirarchial Inheritance
* Hybrid Inheritance
* Multipath Inheritance

4. Inline function is a function in C++ that is expanded in line when it is called. When the inline function is called, whole code of the inline function gets inserted or substituted at the point of inline function call. This substitution is performed by C++ compiler at compile time. Syntax:

```
inline return_type (parameters)
{
    // Body of function.
}
```

5. A pure virtual function in C++ for which we need not to write any function definition and only we have to declare it.

Shubham 9919103057

6. A class can be also declared to be the friend of some other class. When we create a friend class then all the member functions of the friend class also become the friend of the other class. This requires the condition that the friend becomes class must be first declared or defined.

Friend Functions –

These are special functions which can access the private members of class. They are considered to be a loophole in the OPPs concepts but logical use of them can make them useful in some cases.

7. D – None of the above.

8. C – ::

:: (Scope Resolution operator) can't be overloaded.

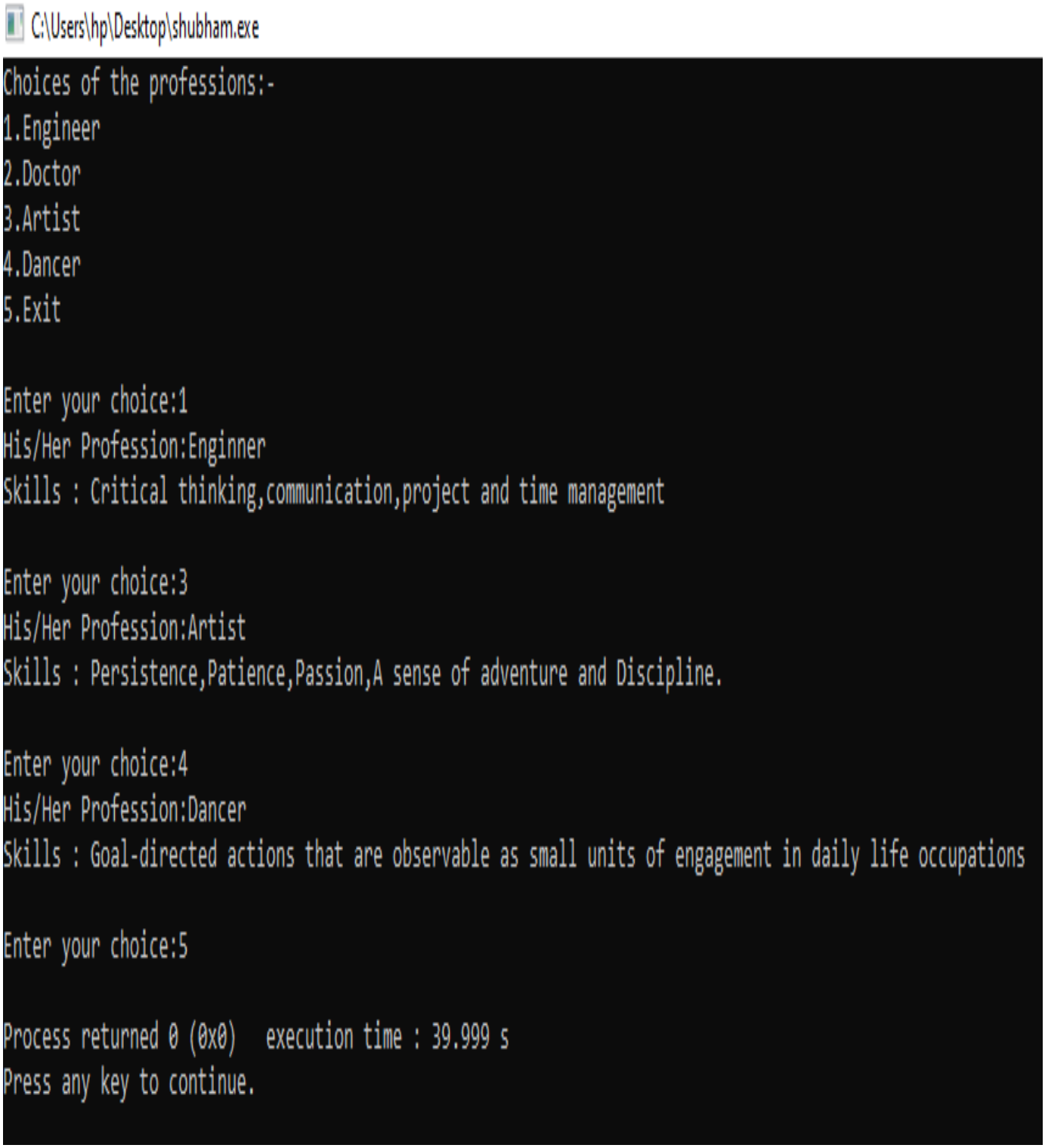
Shubham Garg 9919103057 Batch : F2

Q.9 Write a C++ program to display the skills of a person according to his/her profession using inheritance.

```
#include<iostream>
using namespace std;
class pro
{
   public:
      pro()
      {
         cout<<"Your Profession:";
      }
};
class Artist : public pro
{
   public:
      Artist():pro()
      {
         cout<<"Artist\nSkills : Persistence,Patience,Passion,A sense of adventure and
Discipline.";
      }
};
class Dancer : public pro
{
   public:
      Dancer():pro()
      {
         cout<<"Dancer\nSkills : Goal-directed actions that are observable as small units
of engagement in daily life occupations";
      }
};
class Engineer : public pro
{
   public:
      Engineer():pro()
      {
         cout<<"Enginner\nSkills : Critical thinking,communication,project and time
management";
```

```
        }
};
class Doctor : public pro
{
    public:
        Doctor():pro()
        {
            cout<<"Doctor\nSkills :
Compassion,Understanding,Empathy,Honesty,Competence,Commitment,Humanity and
Courage";
        }
};
int main()
{
    int ch;
    cout<<"Choices of the pros:-"<<endl;
    cout<<"1.Engineer"<<endl;
    cout<<"2.Doctor"<<endl;
    cout<<"3.Artist"<<endl;
    cout<<"4.Dancer"<<endl;
    cout<<"5.Exit";
    while(1)
    {
        cout<<endl<<endl<<"Enter your choice:";
        cin>>ch;
        if(ch==1)
            Engineer e;
        else if(ch==2)
            Doctor d;
        else if(ch==3)
            Artist a;
        else if(ch==4)
            Dancer d;
        else if(ch==5)
            break;
    }
    return 0;
    }
```

Output :

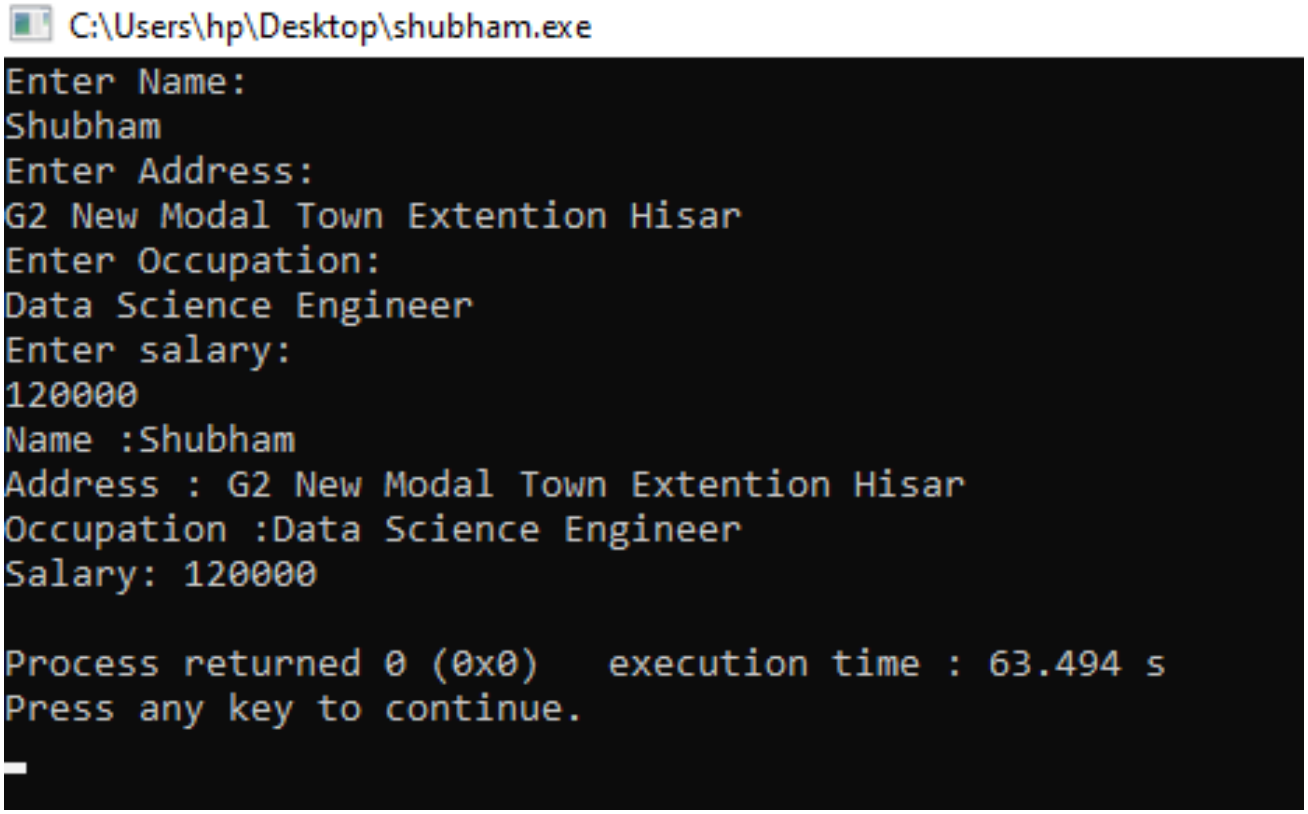

C:\Users\hp\Desktop\shubham.exe
Choices of the professions:-
1.Engineer
2.Doctor
3.Artist
4.Dancer
5.Exit
Enter your choice:1
His/Her Profession:Enginner
Skills : Critical thinking,communication,project and time management
Enter your choice:3
His/Her Profession:Artist
Skills : Persistence,Patience,Passion,A sense of adventure and Discipline.
Enter your choice:4
His/Her Profession:Dancer
Skills : Goal-directed actions that are observable as small units of engagement in daily life occupations
Enter your choice:5
Process returned 0 (0x0) execution time : 39.999 s
Press any key to continue.

Q.10 Write a C++ program to read and print employee information using multiple inheritance.

```
#include<iostream>
#include<cstdio>

using namespace std;

class a1
{
private:
   string name,address;
protected:
   void get()
   {
      cout<<"Enter Name: "<<endl;
      fflush(stdin);
      getline(cin,name);
      cout<<"Enter Address: "<<endl;
      fflush(stdin);
      getline(cin,address);
   }
   void show()
   {
      cout<<"Name :"<<name<<endl;
      cout<<"Address : "<<address<<endl;
   }
};

class a2
{
private:
   string occ;
   int salary;
protected:
   void get()
   {
      cout<<"Enter Occupation:"<<endl;
      fflush(stdin);
      getline(cin,occ);
```

```
        cout<<"Enter salary: "<<endl;
        fflush(stdin);
        cin>>salary;
    }
    void show()
    {
        cout<<"Occupation :"<<occ<<endl;
        cout<<"Salary: "<<salary<<endl;
    }
};

class b:public a1,public a2
{
public:
    b()
    {
        a1::get();
        a2::get();
    }
    void showdata()
    {
        a1::show();
        a2::show();
    }
};

int main()
{
    b emp;
    emp.showdata();
    return 0;
}
```

Output :

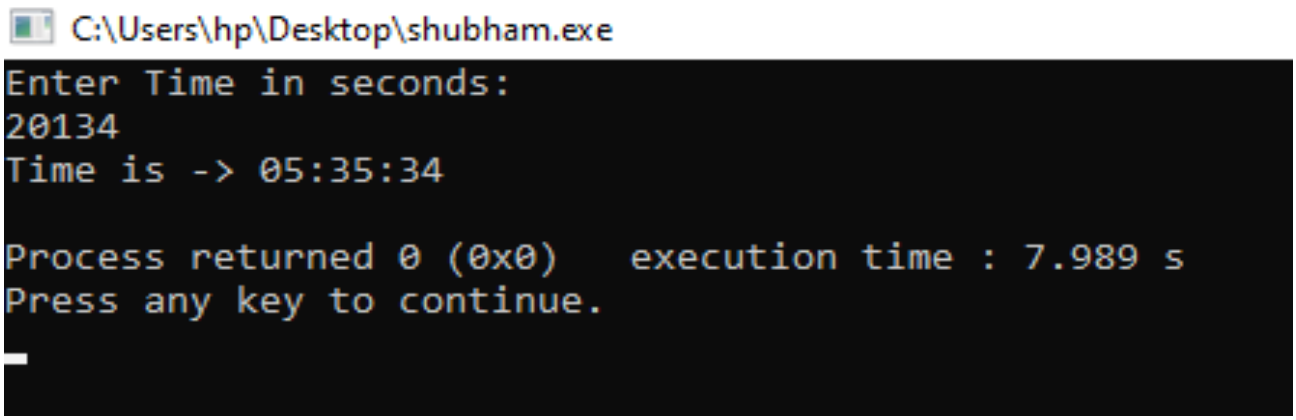


Q.11 Write a C++ program to read time in seconds and convert in time format (HH:MM:SS).

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    cout<<"Enter Time in seconds: "<<endl;
    int n;
    cin>>n;
    int i,j,k,m;
    i=n/3600;
    j=n%3600;
    m=j/60;
    k=j%60;
```

```
    cout<<"Time is ->
"<<setw(2)<<setfill('0')<<i<<":"<<setw(2)<<setfill('0')<<m<<":"<<setw(2)<<setfill('0')
<<k<<endl;
}
```

Output :

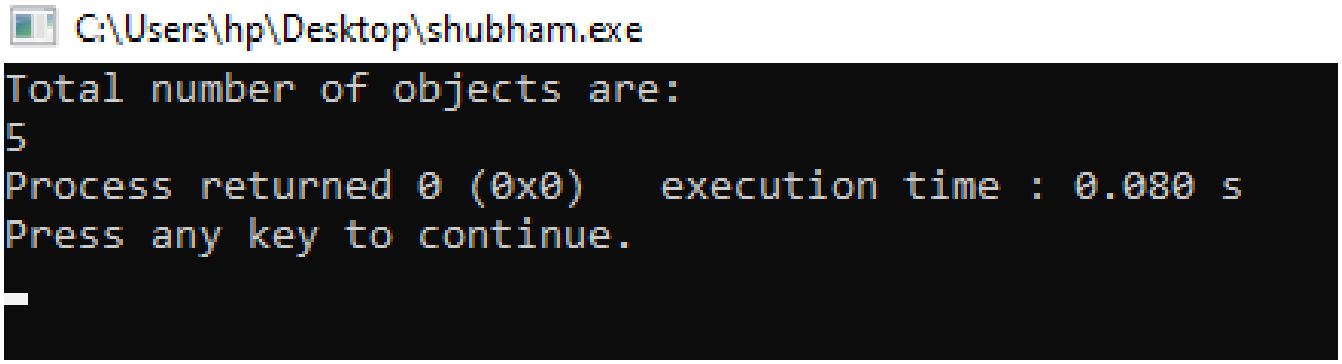


Q.12 Write a C++ program to count the no. of objects created for a class using static member function.

```
#include<iostream>
using namespace std;
class a
{
  public:
  int i;
    static int j;
    a()
    {
       j++;
    }
    static void counter()
    {
       cout<<j;
    }
};
int a::j=0;
int main()
```

```
{
    a a1,a2,a3;
    cout<<"Total number of objects are: "<<endl;
    a::counter();
    return 0;
}
```

Output :

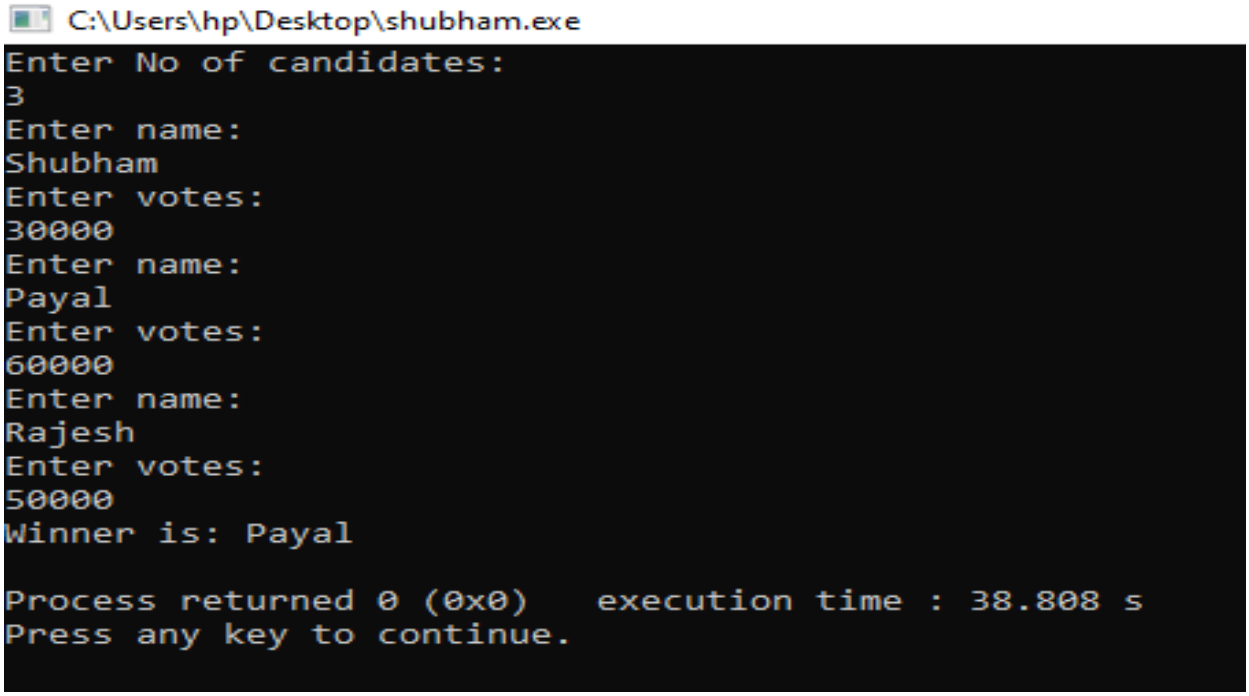


Q.13 Write a C++ program to find the winner of an election based on received votes and no. of candidates.

```
#include<iostream>
#include<cstdio>
using namespace std;
class part
{
    string name;
    int vote;
    public:
    part ()
    {
        cout<<"Enter name: "<<endl;
        fflush(stdin);
        getline(cin,name);
        cout<<"Enter votes: "<<endl;
        cin>>vote;
    }
    static int max;
     static void result(part b[],int n)
    {
```

```
        int i;
        for(i=0;i<n;i++)
        {if(b[i].vote>max)
            max=b[i].vote;
        }
        for(i=0;i<n;i++)
        {
            if(max==b[i].vote)
                break;
        }
        cout<<"Winner is: "<<b[i].name<<endl;
    }
};
int part::max=0;
int main()
{
    cout<<"Enter No of candidates: "<<endl;
    int n;
    cin>>n;
    int i;
    part b[n];
    part::result(b,n);
    return 0;
}
```

Output :

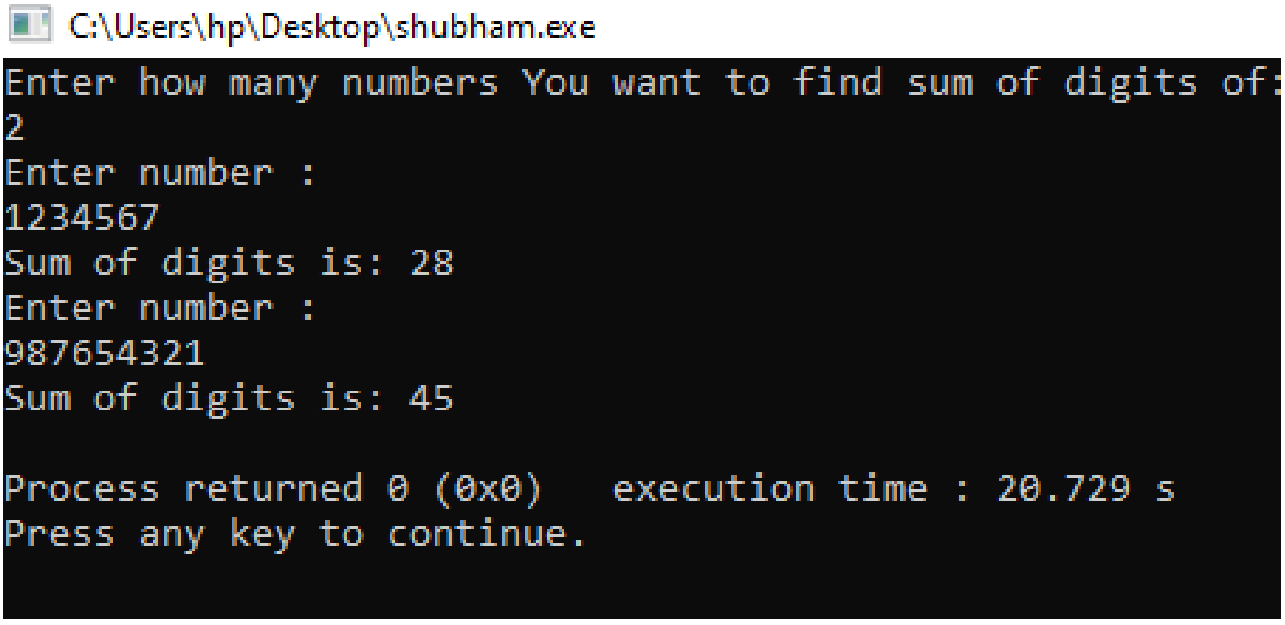


Q.14 Write a C++ program to calculate the sum of the digits of a number.

```
#include<iostream>
using namespace std;
class num
{
  int i;
public:
    void get()
    {
        cout<<"Enter number :"<<endl;
        cin>>i;
    }
    void sum()
    {
        int j,sum=0;
        while(i!=0)
        {
            j=i%10;
            sum=sum+j;
            i=i/10;
        }
        cout<<"Sum of digits is: "<<sum<<endl;
```

```
    }
};
int main()
{
    cout<<"Enter how many numbers You want to find sum of digits of: "<<endl;
    int n;
    cin>>n;
    num b[n];
    int g;
    for(g=0;g<n;g++)
    {
        b[g].get();
        b[g].sum();
    }
    return 0;
}
```

Outputs :



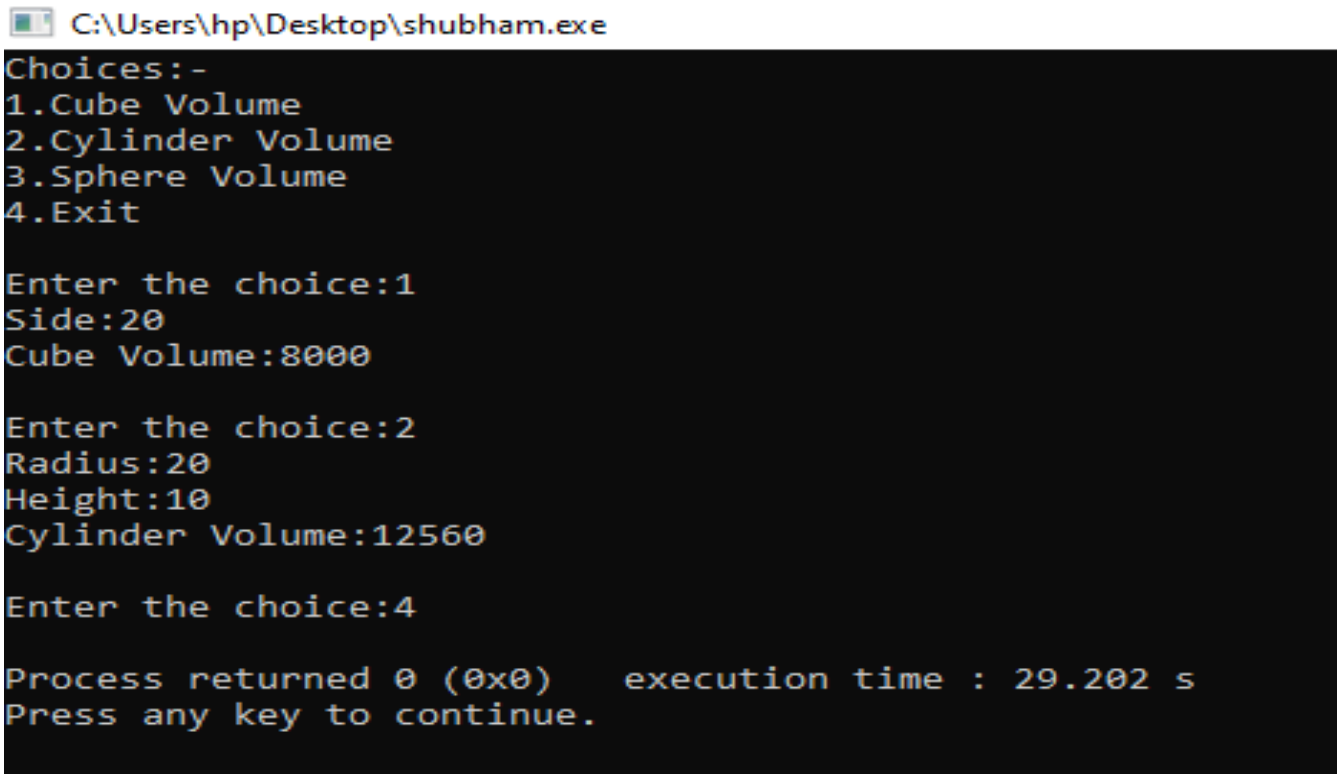
Q.15 Write a C++ program to find volume of cube, cylinder and sphere using function overloading.

```
#include<iostream>
```

```
using namespace std;
class volume
{
    int s,r;
    public:
        void setdata(int x,int y)
        {
            r=x;
            s=y;
        }
        void setdata(int x)
        {
            s=x;
        }
        void getcv()
        {
            cout<<"Cube Volume:"<<s*s*s<<endl;
        }
        void getcyv()
        {
            cout<<"Cylinder Volume:"<<r*r*s*3.14<<endl;
        }
        void getsv()
        {
            cout<<"Sphere Volume:"<<(4*3.14*r*r*r)/3<<endl;
        }
};
int main()
{
    int ch,s,r;
    volume v;
    cout<<"Choices:-\n";
    cout<<"1.Cube Volume\n";
    cout<<"2.Cylinder Volume\n";
    cout<<"3.Sphere Volume\n";
    cout<<"4.Exit\n";
    while(1)
    {
        cout<<"\nEnter the choice:";
        cin>>ch;
```

```
        switch(ch)
        {
            case 1: cout<<"Side:";
                    cin>>s;
                    v.setdata(s);
                    v.getcv();
                    break;
            case 2: cout<<"Radius:";
                    cin>>r;
                    cout<<"Height:";
                    cin>>s;
                    v.setdata(r,s);
                    v.getcyv();
                    break;
            case 3: cout<<"Radius:";
                    cin>>r;
                    v.setdata(s);
                    v.getsv();
                    break;
        }
        if(ch==4)
            break;
    }
    return 0;
}
```

Output :



Q.16 Write a C++ program to add two objects using binary plus (+) operator overloading.

```
#include<iostream>
using namespace std;
class sum
{
   int a;
public:
   sum(int s)
   {
      a=s;
   }
   int operator +(sum const &d)
   {
      int add;
      add=a+d.a;
      return add;
   }
};
int main()
```

```
{
    sum w(2),x(3),y();
    int a=w+x;
    cout<<"Sum is : "<<a<<endl;
    return 0;
}
```

Output :