-3]Q1

```
#include<iostream>
using namespace std;
class volume
{  int v;
public:
void vol(int r,int h)
{  int v=3.14*r*r*h;
cout<<"volume of cylinder:"<<v;
}  void vol(int l)
{   int v=l*l*l;
cout<<"volume of cube:"<<v;
}  void vol(float r)
{  float v=(4/3)*3.14*r*r;
cout<<"volume of sphere:"<<v;    }   };
int main()
{  int r,h,l;
float radius;
volume v1;
cout<<"enter radius and height of cylinder:";
cin>>r>>h;
v1.vol(r,h);
cout<<"\nenter length of cube:";
cin>>l;
v1.vol(l);
cout<<"\nenter radius of sphere:";
cin>>radius;
v1.vol(radius);  }
```

3]Q2) write a c++ program with student as abstract class and create derive class

```
#include <iostream>        #include <string>
using namespace std;
class Student {           protected:
int rollno;
string name;
public:
Student(int rollno, string name) : rollno(rollno), name(name) {}
virtual void display() = 0;
virtual ~Student() {}
};  class Engineering : public Student {
public:
Engineering(int rollno, string name) : Student(rollno, name) {}
void display() override {
cout << "Engineering Student - Roll No: " << rollno << ", Name: " <<
name << endl;
}  };
class Medicine : public Student {
public:
Medicine(int rollno, string name) : Student(rollno, name) {}
void display() override {
cout << "Medicine Student - Roll No: " << rollno << ", Name: " << name
<< endl;
}   };
class Science : public Student {
public:
Science(int rollno, string name) : Student(rollno, name) {}
void display() override {
cout << "Science Student - Roll No: " << rollno << ", Name: " << name
<< endl;
}    };
```

```
int main() {   const int size = 3;
Student* students[size];
for (int i = 0; i < size; ++i) {
int rollno;
string name;
int choice;
cout << "Enter details for student " << i+1 << ":" << endl;
cout << "Roll No: ";
cin >> rollno;
cout << "Name: ";
cin.ignore();
getline(cin, name);
cout << "Choose the stream (1 for Engineering, 2 for Medicine, 3 for
Science): ";
cin >> choice;
switch (choice) {          case 1:
students[i] = new Engineering(rollno, name);
break;
case 2:
students[i] = new Medicine(rollno, name);
break;
case 3:
students[i] = new Science(rollno, name);
break;
default:
cout << "Invalid choice." << endl;    }   }
cout << "\nDetails of all students:" << endl;
for (int i = 0; i < size; ++i) {
students[i]->display();  }
for (int i = 0; i < size; ++i) {
delete students[i];   }      return 0;  }
```