1. Friend Function:
    a. Write down a class named **Vector** and provide appropriate function to make the given ***main*** function executable.

```
class Vector{
private:
    double x,y,z;
public:
    Vector(){}
    Vector(double x,double y,double z){
        this->x=x;
        this->y=y;
        this->z=z;
    }

};

int main(){

    Vector v,mult;


    // Overload >> operator to take Vector input
    cin>>v;


    // Perform scalar multiplication of v and store it into mult
    // v should be unchanged after multiplication.
    mult=5*v;

    // Overload << operator to perform Vector output.
    cout<<v;
    cout<<mult;
    return 0;
}
```

2. Inheritance:

   Write down the following classes as described. Use virtual base class where needed.

   a. Write a class named Cricketer.

Derives:	
none	none
Private Members Variable	
● matchPlayed	an integer
● *name*	an array of characters.
● *age*	an integer
Public Member Function	
● Cricketer(char *s,int a,int m) ○ Constructor	
● char *getName()	

   b. Write a class named Batsman. It should derive Cricketer class as its parent.

Derives:	
Cricketer	public
Private Members Variable	
● runScored	
Public Member Function	

● Batsman(char *s,int age,int matchPlayed,int runScorred) ○ Constructor
● double computeBattingAverage() ○ This functions divides the runScored by matchPlayed.

c. Write a class named Bowler. It should derive Cricketer class as its parent.

Derives:	
Cricketer	public
Private Members Variable	
● wicketsTaken	an integer
Public Member Function	
● double computeWicketAverage()	This functions divides the wicketsTaken by matchPlayed.

d. Write a class named AllRounder.It should derive both Batsman and Bowler class as its parent.

Derives:	
Batsman	public
Bowler	public
Public Member Function	
● AllRounder(char *s,int age,int matchPlayed,int runScorred,int runGiven) ○ Constructor	

Sample Main:

```
int main(){
        AllRounder a("Shakib",28,200,5000,400);
        cout<<a.getName()<<" "<<a.computeAverageRunPerMatch()<<" "<<a.computeBattingAverage()<<endl;
        return 0;
}
```