Q1.

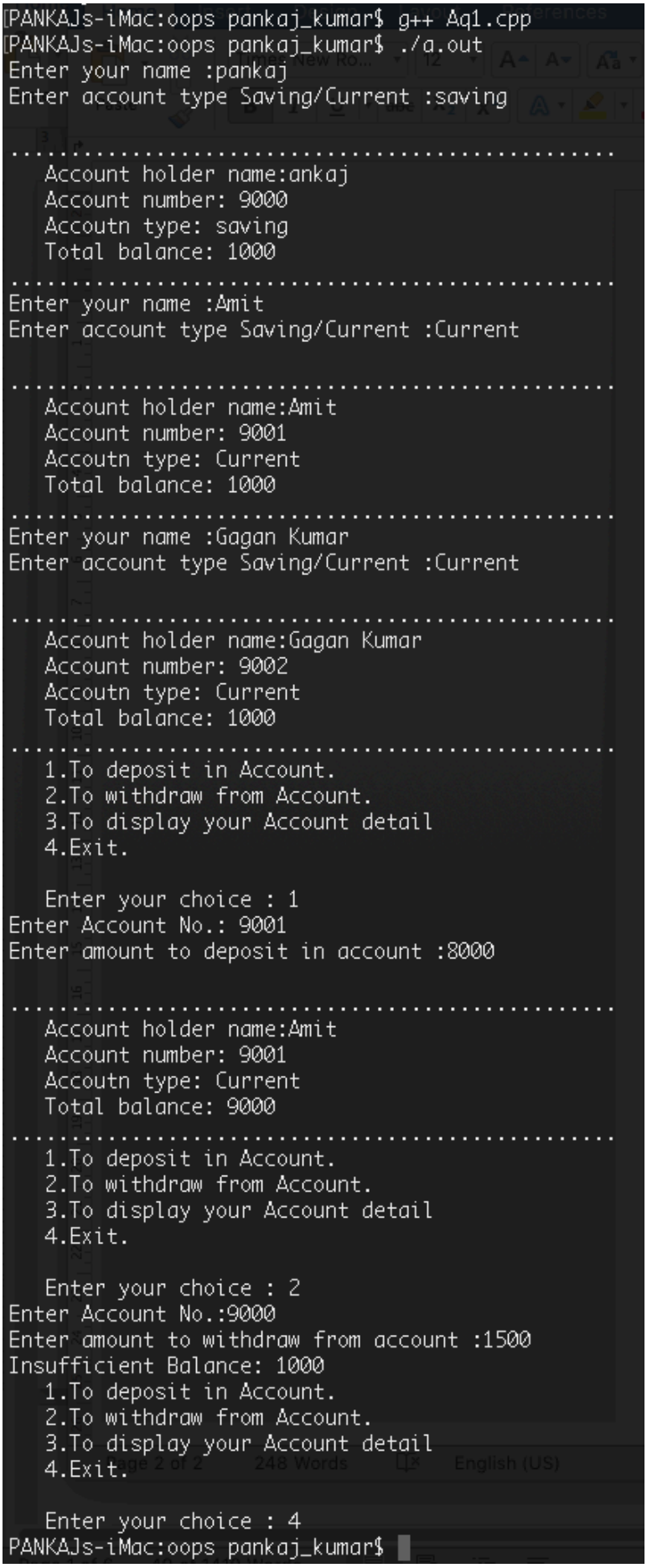
Write a class representing bank accounts. The class should have following data members: Customer Name, Account Number, Type of Account (Savings/Current etc.), Account Balance. The class should allow basic operations like creating a new account, deposit an amount, withdraw money after checking the balance, display account details.

Sol.

```
#include<iostream>

using namespace std;

class Bank_Account{
       char c_name[30];
       int c_acc_no;
       char acc_type[10];
       long balance;
public:
       Bank_Account(){
       balance=1000;
       }
       void create_acc();
       void deposit_amount();
       void withdraw();
       void display();
       int check_acc_no(){
       return c_acc_no;
       }
};
       int acc=9000;
int check_acc_no(int);

int main(){
       int c,i;
       Bank_Account acc[3];
       for(int i=0;i<3;i++){
       acc[i].create_acc();
       }
       do{
       cout<<"   1.To deposit in Account."<<endl;
       cout<<"   2.To withdraw from Account."<<endl;
       cout<<"   3.To display your Account detail"<<endl;
       cout<<"   4.Exit."<<endl;
       cout<<endl<<"   Enter your choice : ";
       cin>>c;
       switch(c){
       case 1:
       cout<<"Enter Account No.: ";
       cin>>c;
       for(i=0;i<3;i++){
       if(acc[i].check_acc_no()==c)
       break;
       }
       acc[i].deposit_amount();
       break;
       case 2:
       cout<<"Enter Account No.:";
       cin>>c;
       for(i=0;i<3;i++){
        if(acc[i].check_acc_no()==c)
        break;
        }
       acc[i].withdraw();
       break;
       case 3:
       cout<<"Enter Account No.: ";
       cin>>c;
       for(i=0;i<3;i++){
        if(acc[i].check_acc_no()==c)
        break;
        }
       acc[i].display();
       break;
```

```
        case 4:
        exit(0);
        }
        }while(true);
        return 0;
}



void Bank_Account::create_acc(){
        cout<<"Enter your name : ";
        cin.ignore();
        cin.getline(c_name,30);
        cout<<"Enter account type Saving/Current :";
        cin>>acc_type;
        c_acc_no=acc;
        acc++;
        display();
}

void Bank_Account::deposit_amount(){
        double amount;
        cout<<"Enter amount to deposit in account :";
        cin>>amount;
        balance+=amount;
        display();
}

void Bank_Account::withdraw(){
        double amount;
        cout<<"Enter amount to withdraw from account :";
        cin>>amount;
        if(amount>balance){
        cout<<"Insufficient Balance: ";
        cout<<balance<<endl;
        return;
        }
        balance-=amount;
        display();
}

void Bank_Account::display(){
        cout<<endl<<"................................................."<<endl;
        cout<<"   Account holder name: "<<c_name<<endl;
        cout<<"   Account number: "<<c_acc_no<<endl;
        cout<<"   Accoutn type: "<<acc_type<<endl;
        cout<<"   Total balance: "<<balance<<endl;
        cout<<"................................................."<<endl;
}
```

OUTPUT :



```
[PANKAJs-iMac:oops pankaj_kumar$ g++ Aq1.cpp
[PANKAJs-iMac:oops pankaj_kumar$ ./a.out
Enter your name :pankaj
Enter account type Saving/Current :saving

................................................
   Account holder name:ankaj
   Account number: 9000
   Accoutn type: saving
   Total balance: 1000
................................................
Enter your name :Amit
Enter account type Saving/Current :Current

................................................
   Account holder name:Amit
   Account number: 9001
   Accoutn type: Current
   Total balance: 1000
................................................
Enter your name :Gagan Kumar
Enter account type Saving/Current :Current

................................................
   Account holder name:Gagan Kumar
   Account number: 9002
   Accoutn type: Current
   Total balance: 1000
................................................
   1.To deposit in Account.
   2.To withdraw from Account.
   3.To display your Account detail
   4.Exit.

   Enter your choice : 1
Enter Account No.: 9001
Enter amount to deposit in account :8000

................................................
   Account holder name:Amit
   Account number: 9001
   Accoutn type: Current
   Total balance: 9000
................................................
   1.To deposit in Account.
   2.To withdraw from Account.
   3.To display your Account detail
   4.Exit.

   Enter your choice : 2
Enter Account No.:9000
Enter amount to withdraw from account :1500
Insufficient Balance: 1000
   1.To deposit in Account.
   2.To withdraw from Account.
   3.To display your Account detail
   4.Exit.

   Enter your choice : 4
PANKAJs-iMac:oops pankaj_kumar$
```

Q2.

Write an Employee class representing an Employee with an Organization. The class has following data: Employee ID, Employee Name, Date of Birth, Salary of the employee. Write functions to add employee details and display employee details**.**

Sol.

```
#include<iostream>

using namespace std;

class Employee{
	int employee_id;
	char employee_name[30];
	char D_O_B[12];
	double salary;
public:

	void add_employee_detail();
	void display_employee_detail();
};

int id=2100;

int main(){
	Employee emp[3];
	for(int i=0;i<3;i++){
	emp[i].add_employee_detail();
	}
	for(int j=0;j<3;j++){
	emp[j].display_employee_detail();
	}
}

void Employee::display_employee_detail(){
	cout<<endl<<"..............................."<<endl;
	cout<<endl<< "  Name of the Employee: "<<employee_name<<endl;
	cout<<"   Employee ID: "<<employee_id<<endl;
	cout<<"   Employee DOB: "<<D_O_B<<endl;
	cout<<"   Salary : "<<salary<<endl;
	cout<<"..............................."<<endl;
	}

void Employee::add_employee_detail(){
	cout<<"Name of the Employee: ";
	cin.ignore();
	cin.getline(employee_name,30);
	cout<<"Enter date of birth in the format of dd/mm/yyyy: ";
	cin.ignore();
	cin.getline(D_O_B,12);
	 cout<<"Enter the salary of the employee: ";
	 cin>>salary;
	employee_id=id;
	id++;
}
```

OUTPUT :

```
PANKAJs-iMac:oops pankaj_kumar$ g++ Aq2.cpp
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
Name of the Employee: Pankaj Kumar
Enter date of birth in the format of dd/mm/yyyy: 05/01/1994
Enter the salary of the employee: 35000
Name of the Employee: Amit
Enter date of birth in the format of dd/mm/yyyy: 05/05/1994
Enter the salary of the employee: 40000
Name of the Employee: Gagan Kumar
Enter date of birth in the format of dd/mm/yyyy: 26/12/1995
Enter the salary of the employee: 38000

.............................

  Name of the Employee: ankaj Kumar
   Employee ID: 2100
   Employee DOB: 5/01/1994
   Salary : 35000
.............................

.............................

  Name of the Employee: Amit
   Employee ID: 2101
   Employee DOB: 5/05/1994
   Salary : 40000
.............................

.............................

  Name of the Employee: Gagan Kumar
   Employee ID: 2102
   Employee DOB: 6/12/1995
   Salary : 38000
.............................
PANKAJs-iMac:oops pankaj_kumar$
```

Q3.

Passing by reference is also an effective way to allow a function to return more than one value. Write a program that returns the previous and next numbers of the first parameter passed using the concept of pass by reference.

Sol.

```
#include <iostream>

using namespace std;

void prevnext (int &x, int& prev, int& next)
       {
       prev = x-1;
       next = x+1;
}

int main (){
       int a, b, c;
       cout<<endl<<"    ............................";
       cout<<endl<<"     Enter a Number: ";
       cin>>a;
       prevnext(a, b, c);
       cout <<endl<<"    Previous=" << b << ",  Next=" << c;
       cout<<endl<<"    ............................"<<endl;
       return 0;
}
```

OUTPUT :

```
PANKAJs-iMac:oops pankaj_kumar$ g++ Aq3.cpp
PANKAJs-iMac:oops pankaj_kumar$ ./a.out

    ..............................
     Enter a Number: 99

    Previous=98,  Next=100
    ..............................
PANKAJs-iMac:oops pankaj_kumar$
```

Q4.

Write a program to sort two numbers using call by reference. Smallest number should be output first.

Sol.

```
#include <iostream>

using namespace std;

void sort (int& p, int& n){
       int temp;
       if(p<n)
       temp=0;
       else
       {
       temp=p;
       p=n;
       n=temp;
       }
}

int main ()
{
       int a,b;
       cout<<endl<<"..............................";
       cout<<endl<<"     Enter first no.: ";
       cin>>a;
       cout<<endl<<"     Enter second no.: ";
       cin>>b;
       cout<<endl<<"..............................";
       sort(a,b);
       cout <<endl<<"     After sorting : " << a <<", "<< b;
       cout<<endl<<"..............................."<<endl;
  return 0;
}
```

OUTPUT :

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out

    ..............................
     Enter a Number: 99

    Previous=98,  Next=100
    ..............................
PANKAJs-iMac:oops pankaj_kumar$ g++ Aq4.cpp
PANKAJs-iMac:oops pankaj_kumar$ ./a.out

..............................
     Enter first no.: 56

     Enter second no.: 45

..............................
     After sorting : 45, 56
..............................
```

Q6.

Create two classes KILOS and POUNDS which store the value of weights. KILOS store weight in kilograms and grams and POUNDS in pounds and ounces. Write a program using friend function to add weight of a KILOS object to weight of POUNDS object. Store the result as POUNDS object. (Use 1 pound = 16 ounces, 1 ounce = 28 grams)

Sol.

```
#include<iostream>

using namespace std;

	class KILOS;

	class POUNDS{
		int pounds;
		int ounces;
	public:
		void getdata();
		void display();
	friend POUNDS add ( POUNDS & , KILOS & );
	};

	class KILOS{
		int kg;
		int grams;
	public :
		void getdata();
	friend POUNDS add ( POUNDS & , KILOS &);
	};

	POUNDS add (POUNDS &x ,KILOS &y){
		POUNDS p;
		p.ounces=16*x.pounds + x.ounces;
		p.ounces+= (1000*y.kg + y.grams )/ 28;
		p.pounds = p.ounces /16;
		p.ounces = p.ounces % 16;
		return p;
	}


	int main(){
		KILOS k;
		POUNDS p;
		k.getdata();
		p.getdata();
		POUNDS total;
		total = add(p,k);
		total.display();
	}

	void KILOS::getdata(){
			cout<<endl<<".....................................";
			cout<<endl<<"     Enter the values in Kilogram: ";
			cin>>kg;
			cout<<endl<<"     Enter the value of grams: ";
			cin>> grams;
			cout<<"....................................."<<endl;
	}

	void POUNDS::getdata (){
			cout<<endl<<".....................................";
			cout<<endl<<"     Enter the value in pounds: ";
			cin>>pounds;
			cout<<endl<<"     Enter the value in ounces: ";
			cin>>ounces;
			cout<<".....................................";
		}
	  void POUNDS::display(){
			cout<<endl<<"....................................."<<endl;
			cout<<"   Total in Pounds: "<<pounds<<" and Ounces: "<<ounces;
```

```
                cout<<endl<<"....................................."<<endl;
        }
```

OUTPUT :

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out

.....................................
     Enter the values in Kilogram: 15

     Enter the value of grams: 900
.....................................

.....................................
     Enter the value in pounds: 14

     Enter the value in ounces: 12
.....................................
.....................................
   Total in Pounds: 50 and Ounces: 3
.....................................
```

Q7.
Write an overloaded volume function to calculate the volume of a cube (side*side*side), a cuboid (l*b*h) and a cylinder.

Sol.

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

using namespace std;

                void volume(float l,float b, float h);
                void volume(float r,float h);
                void volume(float l);

        int main(){
                int c;
                float l,b,h,r;
                do{
                        cout<<endl<<"....................................."<<endl;
                        cout<<endl<<"   1.To calculate volume of cube."<<endl;
                        cout<<"   2.To calculate volume of cuboid."<<endl;
                        cout<<"   3.To calculate volume of cylinder."<<endl;
                        cout<<"   4.Exit."<<endl;
                        cout<<endl<<"....................................."<<endl;
                        cout<<"     Enter your choice: ";
                        cin>>c;
                        switch(c){
                                case 1:
                                cout<<"   Enter the side of cube: ";
                                cin>>l;
                                volume(l);
                                break;
                                case 2:
                                cout<<"   Enter the length of cuboid: ";
                                cin>>l;
                                cout<<"   Enter the breath of cuboid: ";
                                cin>>b;
                                cout<<"   Enter the height of cuboid: ";
                                cin>>h;
                                volume(l,b,h);
                                break;
                                case 3:
                                cout<<"   Enter the radius of cylinder: ";
                                cin>>r;
                                cout<<"   Enter the height of cylinder: ";
                                cin>>h;
                                volume(r,h);
```

```
                        break;
                        case 4:
                        exit(0);
                }
        }while(true);
        return 0;
}

void volume(float l,float b,float h){
        float volume;
        volume=l*b*h;
        cout<<endl<<"......................................"<<endl;
        cout<<endl<<"     Volume of cuboid is: "<<volume<<endl;
        cout<<endl<<"......................................"<<endl;
}

void volume(float r,float h){
        float volume;
        volume=(3.14*r*r*h);
        cout<<endl<<"......................................"<<endl;
        cout<<endl<<"      Volume of cylinder is: "<<volume<<endl;
        cout<<endl<<"......................................"<<endl;
}

void volume(float l){
        float volume;
        volume=l*l*l;
        cout<<endl<<"......................................"<<endl;
        cout<<endl<<"       Volume of cube: "<<volume<<endl;
        cout<<endl<<"......................................"<<endl;
}
```

OUTPUT :

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out

....................................

   1.To calculate volume of cube.
   2.To calculate volume of cuboid.
   3.To calculate volume of cylinder.
   4.Exit.

....................................
     Enter your choice: 1
   Enter the side of cube: 7

....................................

       Volume of cube: 343

....................................

....................................

   1.To calculate volume of cube.
   2.To calculate volume of cuboid.
   3.To calculate volume of cylinder.
   4.Exit.

....................................
     Enter your choice: 2
   Enter the length of cuboid: 4
   Enter the breath of cuboid: 5
   Enter the height of cuboid: 9

....................................

     Volume of cuboid is: 180

....................................

....................................
```

```
   1.To calculate volume of cube.
   2.To calculate volume of cuboid.
   3.To calculate volume of cylinder.
   4.Exit.

.....................................
     Enter your choice: 3
   Enter the radius of cylinder: 4
   Enter the height of cylinder: 6

.....................................

      Volume of cylinder is: 301.44

.....................................

.....................................

   1.To calculate volume of cube.
   2.To calculate volume of cuboid.
   3.To calculate volume of cylinder.
   4.Exit.

.....................................
     Enter your choice: 4
```

Q8.

Write a program to copy data from one int array to another. Use the concept of new and delete to allocate and de-allocate memory for the arrays.

Sol.

```
#include <iostream>

 using namespace std;

int main()
{
    cout << "Enter a positive integer: ";
    int length,value,i;
    cin >> length;
    int *array1 = new int[length];
    int *array2 = new int[length];
    cout << "I just allocated an array of integers of length " << length <<endl<<endl;
    for(i=0;i<length;i++){
    cout<<"  array1["<<i<<"]=";
    cin>>value;
    array1[i]=value;
    array2[i]=array1[i];
    }
    cout<<endl<<"..................."<<endl<<endl;
    for(int j=0;j<length;j++){
    cout<<"  array2["<<j<<"]="<<array2[j]<<endl;
    }
    cout<<endl;
       delete[] array1;
       delete[] array2;
    array1 = 0;
    array2 = 0;

    return 0;
}
```

OUTPUT :

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
Enter a positive integer: 5
I just allocated an array of integers of length 5

  array1[0]=12
  array1[1]=13
  array1[2]=14
  array1[3]=15
  array1[4]=16

...................

  array2[0]=12
  array2[1]=13
  array2[2]=14
  array2[3]=15
  array2[4]=16
```

Q9.

An electricity board charges the following rates to domestic users to discourage large consumption of energy:
For the first 100 units - 60P per unit
For next 200 units - 80P per unit
Beyond 300 units - 90P per unit
All users are charged a minimum of Rs. 50.00. If the total amount is more than Rs. 300.00 then an additional surcharge of 15% is added.
Write a program to read the names of users and number of units consumed and print out the charges with names.

Sol.

```
#include<iostream>

using namespace std;
	struct ele_board{
		char name[30];
		double unit;
		double charges;
		};
double bill(double);

int main(void){
	struct ele_board record[5];
	for(int i=0;i<5;i++){
	cout<<i+1<<" Enter name: ";
	cin>>record[i].name;
	cout<<i+1<<" Enter unit ";
	cin>>record[i].unit;
	record[i].charges=bill(record[i].unit);
	}
	cout<<"  S.No"<<"  Name     "<<"  Unit  "<<"  Amount Charges   "<<endl;
	for(int j=0;j<5;j++){
	cout<<"   "<<j+1<<"  "<<record[j].name<<"       "<<record[j].unit<<"    "<<record
[j].charges<<endl;
	}
}

double bill(double unit){
	double amount=0;
	for(int i=1;i<=unit;i++){
	if(i<=100)
	{
	amount+=0.6;
	}
	else if(i<=300)
	{
	amount+=0.8;
	}
```

```
        else if(i>300)
        {
        amount+=0.9;
        }
        }
        if(amount<50)
        {
        return 50;
        }
        if (amount>300)
        {
        amount+=(amount*0.15);
        }
        return amount;
}
```

OUTPUT :

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
1 Enter name: Pankaj
1 Enter unit 500
2 Enter name: Gagan
2 Enter unit 400
3 Enter name: Yogi
3 Enter unit 600
4 Enter name: Amit
4 Enter unit 300
5 Enter name: Hitesh
5 Enter unit 80
  S.No  Name       Unit    Amount Charges
   1  Pankaj       500    460
   2  Gagan        400    356.5
   3  Yogi         600    563.5
   4  Amit         300    220
   5  Hitesh       80     50
```

Q10.

Sol.

```
#include<iostream>

using namespace std;

  class DB;

  class DM{
      int metre;
      int cm;
  public:
    void getdata();
    void display();
  friend DM add ( DM & , DB & );
  };

  class DB{
      int feet;
      int inch;
  public :
    void getdata();
  friend DM add ( DM & , DB &);
  };

  DM add (DM &x ,DB &y){
    DM p;
            p.cm  = 100*x.metre + x.cm;
            p.cm+= (30*y.feet)+(3*y.inch);
            p.metre = p.cm /100;
            p.cm = p.cm % 100;
```

```
    return p;
  }


int main(){
  DB k;
  DM p;
  k.getdata();
  p.getdata();
  DM total;
  total = add(p,k);
  total.display();
}

void DM::getdata(){
                    cout<<endl<<".....................................";
                    cout<<endl<<"     Enter the values in Metre: ";
                    cin>>metre;
                    cout<<endl<<"     Enter the value of Centimetre: ";
                    cin>> cm;
                    cout<<"....................................."<<endl;
                }

void DB::getdata (){
                    cout<<endl<<".....................................";
                    cout<<endl<<"     Enter the value in Feet: ";
                    cin>>feet;
                    cout<<endl<<"     Enter the value in Inch: ";
                    cin>>inch;
                    cout<<".....................................";
                 }
void DM::display(){
                    cout<<endl<<".....................................";
                    cout<<"   Total in Metre: "<<metre<<" and Centimetre: "<<cm;
                    cout<<endl<<".....................................";
                }
```

OUTPUT :

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out

.....................................
     Enter the value in Feet: 31

     Enter the value in Inch: 9
.....................................
.....................................
     Enter the values in Metre: 15

     Enter the value of Centimetre: 90
.....................................

.....................................
   Total in Metre: 25 and Centimetre: 47
.....................................
```

Q9.
Write a box class with class member's length, breadth, height and a function to calculate the volume of the box. Constructor should be passed arguments for length, breadth, height. Also define a second constructor which takes default height value of 10.

Sol-

```
#include <iostream>

using namespace std;

class Box{
                float length;
                float breath;
                float height;
        public:
                Box(float l,float b,float h){
                        length=l;
                        breath=b;
                        height=h;
                }

                Box(float l,float b){
                        length=l;
                        breath=b;
                        height=10;
                }

                float volume(){
                        return (length*breath*height);
                }
};

int main(){
        float vol1,vol2;
        float l,b,h;
        cout<<"  Enter the length of box : ";
        cin>>l;
        cout<<"  Enter the breath of box : ";
        cin>>b;
        cout<<"  Enter the height of box : ";
        cin>>h;

        Box B1(l,b,h),B2(l,b);
        vol1=B1.volume();
        vol2=B2.volume();
        cout<<"  Volume of Box(B1) : "<<vol1<<endl;
        cout<<"  Volume of Box(B2) whose height is 10 : "<<vol2<<endl;

        return 0;
}
```

OUTPUT –

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Enter the length of box : 5
  Enter the breath of box : 5
  Enter the height of box : 5
  Volume of Box(B1) : 125
  Volume of Box(B2) whose height is 10 : 250
PANKAJs-iMac:oops pankaj_kumar$
```

Q10.
Write a class with a "int *" member function whose constructor takes a int argument and allocates memory using "new" and destructor frees that memory. (Note : Copy constructor should also be defined for this class.)

Sol-

```
#include <iostream>

using namespace std;

class Length {
        int *ptr;
public:
        Length( int len );
        Length(const Length &obj);
        ~Length();
        int getLength();
};

Length::Length( int len){
        cout<<"  Normal constructor allocation ptr."<<endl;
        ptr = new int;
        *ptr = len;
}

Length::Length( const Length &obj){
        cout<<"  Copy constructor allocating ptr."<<endl;
        ptr = new int;
        *ptr = *obj.ptr;
}

Length::~Length(){
        cout<<"  Freeing memory."<<endl;
        delete ptr;
}

int Length::getLength(){
        return *ptr;
}

void display(Length obj){
        cout<<"  Length : "<<obj.getLength()<<endl;
}

int main(){
        Length line1(20);
        Length line2 = line1; //It calls copy constructor
        Length line3(10);
        display(line1);
        display(line2);
        display(line3);
        return 0;
}
```

OUTPUT-

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Normal constructor allocation ptr.
  Copy constructor allocating ptr.
  Normal constructor allocation ptr.
  Copy constructor allocating ptr.
  Length : 20
  Freeing memory.
  Copy constructor allocating ptr.
  Length : 20
  Freeing memory.
  Copy constructor allocating ptr.
  Length : 10
  Freeing memory.
  Freeing memory.
  Freeing memory.
  Freeing memory.
PANKAJs-iMac:oops pankaj_kumar$
```

Q11.
Write a class with a simple integer as class member and a member function that returns that value. There should be two overloaded constructors one which takes an integer as argument and the other taking a “char *” as argument.(Note : Can use atoi function to convert “char *” to int.)

Sol-

```
#include <iostream>
#include <cstdlib>
#include <cstring>

using namespace std;

class Integer{
	int number;
public:
	Integer( int n ){
		number = n;
	}

	Integer( const char *s ){
		char *str;
		int len = strlen(s);
		str = new char[len + 1];
		strcpy(str,s);
		number = atoi(str);
	}

	int value(){
		return number;
	}
};

int main(){
	int no1,no2;
	Integer No(90);
	Integer char_No("12345");
	no1 = No.value();
	cout<<"  No1 is : "<<no1<<endl;
	no2 = char_No.value();
	cout<<"  No2 is (char) : "<<no2<<endl;
	return 0;
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  No1 is : 90
  No2 is (char) : 12345
PANKAJs-iMac:oops pankaj_kumar$
```

Q12.
Extend assignment 2 to have employees data entered through constructor(s). Write default and copy constructors

Sol-

```
#include <iostream>
#include <cstring>
#include <cstdlib>

using namespace std;

class Employee{
	int employee_id;
	char *employee_name;
	char *D_O_B;
```

```
        double salary;
public:
        Employee();
        Employee(const int id,char *e_name,char * e_dob,double e_s);
        Employee(const Employee &e){
                employee_id = e.employee_id;
                employee_name = e.employee_name;
                D_O_B = e.D_O_B;
                salary = e.salary;
        }

        void add_employee_detail();
        void display_employee_detail();
};

Employee :: Employee(const int id,char *e_name,char * e_dob,double e_s){
                int len_n = strlen(e_name);
                int len_d = strlen(e_dob);
                employee_name = new char [len_n + 1];
                strcpy(employee_name,e_name);
                D_O_B = new char [len_d + 1];
                strcpy(D_O_B,e_dob);
                employee_id = id;
                salary = e_s;
        }

int id=2100;

int main(){
        Employee emp1(2100,"Pankaj Kumar","25/01/1994",35000);
        Employee emp2(2101,"Gagan","11/08/1994",38000);
        Employee emp3(2102,"Nisha","12/01/1993",39000);
        Employee emp4 = emp1;
        emp1.display_employee_detail();
        emp2.display_employee_detail();
        emp3.display_employee_detail();
        emp4.display_employee_detail();
        /*
        for(int i=0;i<3;i++){
        emp[i].add_employee_detail();
        }
        for(int j=0;j<3;j++){
        emp[j].display_employee_detail();
        }
        */
}

void Employee::display_employee_detail(){
        cout<<endl<<"................................."<<endl;
        cout<<endl<< "  Name of the Employee: "<<employee_name<<endl;
        cout<<"   Employee ID: "<<employee_id<<endl;
        cout<<"   Employee DOB: "<<D_O_B<<endl;
        cout<<"   Salary : "<<salary<<endl;
        cout<<"................................."<<endl;
        }
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out

.................................

  Name of the Employee: Pankaj Kumar
   Employee ID: 2100
   Employee DOB: 25/01/1994
   Salary : 35000
.................................

.................................

  Name of the Employee: Gagan
   Employee ID: 2101
   Employee DOB: 11/08/1994
```

```
   Salary : 38000
..............................

..............................

  Name of the Employee: Nisha
   Employee ID: 2102
   Employee DOB: 12/01/1993
   Salary : 39000
..............................

..............................

  Name of the Employee: Pankaj Kumar
   Employee ID: 2100
   Employee DOB: 25/01/1994
   Salary : 35000
..............................
PANKAJs-iMac:oops pankaj_kumar$
```

Q13.

Write a Matrix class with attributes: row, col, dynamically created two-dimensional int array. Define constructors to allocate memory dynamically for the dimensions supplied. Write methods to set, get values of individual matrix elements, display contents of the matrix etc.

Sol-

```
#include <iostream>

using namespace std;

class matrix{
    int r,c;
    int **p;
public:
    matrix (int n1, int n2);
    int elementvalue(int i,int j);
    void getelement(int i,int j,int value);
    void displayelement(int n1,int n2);
};

matrix::matrix(int n1,int n2){
    r=n1;
    c=n2;
    p=new int *[r];
    for(int i=0 ; i<r ; i++){
        p[i] = new int[c];
    }
}

void matrix::getelement(int i,int j,int value){
    p[i][j]=value;
}

void matrix::displayelement(int n1,int n2){
    r=n1;
    c=n2;
    for(int i=0 ; i<n1 ; i++ ){
        cout<<endl<<"  ";
        for(int j=0 ; j<n2 ; j++){
            cout<<p[i][j]<<"\t";
        }
    }
    cout<<endl;
}

int matrix::elementvalue(int i,int j){
    int value;
    value=p[i][j];
    return (value);
}
```

```
int main(){
    int n,m,value;
    cout<<"  Enter size of matrix."<<endl<<"  Rows: ";
    cin>>n;
    cout<<"  Columns: ";
    cin>>m;
    matrix m1(n,m),m2(n,m),m3(n,m);
    cout<<"  Please enter the values: "<<endl;
    for(int i=0 ; i<n ; i++){
        for(int j=0 ; j<m ; j++){
            cout<<"  Matrix1["<<i<<"]["<<j<<"] = ";
            cin>>value;
            m1.getelement(i,j,value);
        }
    }
    cout<<"............................."<<endl;

    for(int i=0 ; i<n ; i++){
        for(int j=0 ; j<m ; j++){
            cout<<"  Matrix2["<<i<<"]["<<j<<"] = ";
            cin>>value;
            m2.getelement(i,j,value);
        }
    }
    cout<<"  Matrix1 : "<<endl;
    m1.displayelement(n,m);
    cout<<endl<<"  Matrix2 : "<<endl;
    m2.displayelement(n,m);

    for(int i=0;i<n;i++)
        for(int j=0;j<m;j++){
            value=m1.elementvalue(i,j);
            m3.getelement(j,i,value);
        }
    cout<<"  Transpose of Matrix1 : "<<endl;
    m3.displayelement(n,m);


    return 0;
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Enter size of matrix.
  Rows: 3
  Columns: 3
  Please enter the values:
  Matrix1[0][0] = 1
  Matrix1[0][1] = 2
  Matrix1[0][2] = 3
  Matrix1[1][0] = 4
  Matrix1[1][1] = 5
  Matrix1[1][2] = 6
  Matrix1[2][0] = 7
  Matrix1[2][1] = 8
  Matrix1[2][2] = 9
.............................
  Matrix2[0][0] = 1
  Matrix2[0][1] = 2
  Matrix2[0][2] = 3
  Matrix2[1][0] = 4
  Matrix2[1][1] = 5
  Matrix2[1][2] = 6
  Matrix2[2][0] = 7
  Matrix2[2][1] = 8
  Matrix2[2][2] = 9
  Matrix1 :

  1    2      3
  4    5      6
  7    8      9
```

```
  Matrix2 :

  1    2      3
  4    5      6
  7    8      9
  Transpose of Matrix1 :

  1    4      7
  2    5      8
  3    6      9
PANKAJs-iMac:oops pankaj_kumar$
```

Q14.
Overload the '+' operator to add two Rational Numbers

Sol-

```
#include <iostream>

using namespace std;

class Rational{
    int x,y;
public:
    Rational(){
        x=0;
        y=0;
    }
    Rational(int a,int b){
        x = a;
        y = b;
    }

    Rational operator +(Rational p){
        Rational temp;
        temp.x = (x*p.y) + (p.x*y);
        temp.y = y*p.y;
        int n,d,r,rn,rd,div=1;
        n = temp.x;
        d = temp.y;
        if( n < d ){
            r = d;
        }
        else
        {
            r = n;
        }
        do{
            rn = temp.x % div;
            rd = temp.y % div;
            if( rn == 0 && rd == 0 ){
                temp.x = temp.x/div;
                temp.y = temp.y/div;
            }
            div++;
        }while(div <= r);

        return temp;
    }
    void display(Rational ,int);
};

void Rational :: display(Rational t,int num){
    int n,d,r,rn,rd,div=1;
    n = t.x;
    d = t.y;
    if( n < d ){
        r = d;
    }
    else
    {
        r = n;
    }
```

```
    do{
        rn = t.x % div;
        rd = t.y % div;
        if( rn == 0 && rd == 0 ){
            t.x = t.x/div;
            t.y = t.y/div;
        }
        div++;
    }while(div <= r);

    if(t.x == t.y || t.y == 1){
        cout<<"   Number "<<num<<" : "<<t.x<<endl;
    }
    else
    {
        cout<<"   Number "<<num<<" : "<<t.x<<"/"<<t.y<<endl;
    }
}

int main(){
    int n1,n2,d1,d2;
    cout<<"   Enter the numerator of number 1 : ";
    cin>>n1;
    cout<<"   Enter the denominator of number 1 :";
    cin>>d1;
    Rational R1(n1,d1);
    R1.display(R1,1);
    cout<<"   Enter the numerator of number 2 : ";
    cin>>n2;
    cout<<"   Enter the denominator of number 2 :";
    cin>>d2;
    Rational R2(n2,d2);
    R2.display(R2,2);
    Rational R3;
    R3 = R1+ R2;
    cout<<"   Final result";
    R3.display(R3,3);
    return 0;
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
   Enter the numerator of number 1 : 2
   Enter the denominator of number 1 :8
   Number 1 : 1/4
   Enter the numerator of number 2 : 4
   Enter the denominator of number 2 :10
   Number 2 : 2/5
   Final result   Number 3 : 13/20
PANKAJs-iMac:oops pankaj_kumar$
```

Q15.

Overload the '*' operator to multiply two matrices

Sol-

```
#include <iostream>

using namespace std;

class Matrix{
    int m[3][3];
public:
    void read(){
        cout<<"  Enter the elemets of 3x3 matrix: \n";
        for(int i = 0 ; i < 3 ; i++ ){
            for(int j = 0 ; j < 3 ; j++ ){
                cout<<"  m["<<i<<"]["<<j<<"] = ";
                cin>>m[i][j];
            }
        }
```

```
    }

    void display(void){
        int i,j;
        for(i=0;i<3;i++){
                cout<<endl<<"  ";
            for(j=0;j<3;j++){
                cout<<m[i][j]<<"\t";
                        }
                }
                cout<<endl;
        }

    Matrix operator * (Matrix m1){
        Matrix temp;
        int i,j,k;
        for(i=0;i<3;i++){
            for(j=0;j<3;j++){
                temp.m[i][j]=0;
                for(k=0;k<3;k++){
                temp.m[i][j]=temp.m[i][j]+m[i][k]*m1.m[k][j];
            }
        }
    }
    return temp;
    }
};



int main(){
    Matrix M1,M2,M3;
    M1.read();
    cout<<endl<<"  The matrix is as follows :"<<endl;
    M1.display();
    M2.read();
    cout<<endl<<"  The matrix is as follows :"<<endl;
    M2.display();
    M3 = M1 * M2;
    cout<<endl<<"  Multipliction of two matrix :";
    M3.display();
    return 0;
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Enter the elemets of 3x3 matrix:
  m[0][0] = 1
  m[0][1] = 2
  m[0][2] = 3
  m[1][0] = 4
  m[1][1] = 5
  m[1][2] = 6
  m[2][0] = 7
  m[2][1] = 8
  m[2][2] = 9

  The matrix is as follows :

  1    2      3
  4    5      6
  7    8      9
  Enter the elemets of 3x3 matrix:
  m[0][0] = 1
  m[0][1] = 2
  m[0][2] = 3
  m[1][0] = 4
  m[1][1] = 5
  m[1][2] = 6
  m[2][0] = 7
  m[2][1] = 8
  m[2][2] = 9
```

```
  The matrix is as follows :

  1    2      3
  4    5      6
  7    8      9

  Multipliction of two matrix :
  30   36     42
  66   81     96
  102  126    150
PANKAJs-iMac:oops pankaj_kumar$
```

Q16.
Overload "<<" and ">>" operators for Employee, Matrix & Rational class defined in above assignments and test the input and output for the same using **cout** and **cin** respectively.

Sol-
16.1

```
#include<iostream>

using namespace std;

class Employee{
       int employee_id;
       char employee_name[30];
       char D_O_B[12];
       double salary;
public:
       friend ostream & operator << (ostream &out, const Employee &E);
       friend istream & operator >> (istream &in, Employee &E);
};

int id=2100;

ostream & operator << (ostream &out, const Employee &E){
       out<<endl<<"................................."<<endl;
       out<<endl<< "  Name of the Employee: "<<E.employee_name<<endl;
       out<<"   Employee ID: "<<E.employee_id<<endl;
       out<<"   Employee DOB: "<<E.D_O_B<<endl;
       out<<"   Salary : "<<E.salary<<endl;
       out<<"................................."<<endl;
}

istream & operator >> (istream &in, Employee &E){
       cout<<"Name of the Employee: ";
       in>>E.employee_name;
       cout<<"Enter date of birth in the format of dd/mm/yyyy: ";
       in>>E.D_O_B;
    cout<<"Enter the salary of the employee: ";
    in>>E.salary;
       E.employee_id=id;
       id++;
}

int main(){
       Employee emp[3];
       for(int i=0;i<3;i++){
       cin>>emp[i];
       }
       for(int j=0;j<3;j++){
       cout<<emp[j];
       }
       return 0;
}
```

16.2

Sol-

```
#include <iostream>

using namespace std;

class matrix{

	int m[3][3];
 public:
	friend ostream & operator << (ostream &out, matrix &M);
	friend istream & operator >> (istream &in, matrix &M);
	friend matrix trans(matrix);
	friend matrix mul(matrix, matrix);
};

ostream & operator << (ostream &out, matrix &M){
	int i,j;
	for(i=0;i<3;i++){
		cout<<endl;
		for(j=0;j<3;j++){
			out<<M.m[i][j]<<"\t";
                                }
			}
			return out;
}

istream & operator >> (istream &in, matrix &M){
	cout<<"Enter the elements of 3*3 matrix:\n";
	int i,j;
	for(i=0;i<3;i++)
		for(j=0;j<3;j++){
			cout<<"m["<<i<<"]["<<j<<"] = ";
			in>>M.m[i][j];
			}
			return in;
}

matrix trans(matrix m1){
	matrix m2;
	int i,j;
	for(i=0;i<3;i++)
		for(j=0;j<3;j++){
			m2.m[i][j]=m1.m[j][i];
		}
	return(m2);
}

matrix mul(matrix m1,matrix m2){
	matrix m3;
	int i,j,k;
    for(i=0;i<3;i++){
	 for(j=0;j<3;j++){
	    m3.m[i][j]=0;
			for(k=0;k<3;k++){

                                m3.m[i][j]=m3.m[i][j]+m1.m[i][k]*m2.m[k][j];
		}
	 }
    }
	return(m3);
}

int main(){
	matrix mat1,mat2,mat3,mat4;
	cin>>mat1;
	cout<<endl<<"The matrix is as follows:";
	cout<<mat1;
	cin>>mat3;
	cout<<endl<<"The matrix is as follows:";
	cout<<mat3;
	mat2=trans(mat1);
	cout<<endl<<"Transposed matrix:";
```

```
        cout<<mat2;
        mat4=mul(mat1,mat3);
        cout<<endl<<"Multiplication on matrixs:";
        cout<<mat4;
        return 0;
}
```

OUTPUT

16.1

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
Name of the Employee: Pankaj
Enter date of birth in the format of dd/mm/yyyy: 01/12/1994
Enter the salary of the employee: 30000
Name of the Employee: Gagan
Enter date of birth in the format of dd/mm/yyyy: 04/10/1995
Enter the salary of the employee: 35000
Name of the Employee: Amit
Enter date of birth in the format of dd/mm/yyyy: 19/01/1994
Enter the salary of the employee: 35000

..............................

  Name of the Employee: Pankaj
   Employee ID: 2100
   Employee DOB: 01/12/1994
   Salary : 30000
..............................

..............................

  Name of the Employee: Gagan
   Employee ID: 2101
   Employee DOB: 04/10/1995
   Salary : 35000
..............................

..............................

  Name of the Employee: Amit
   Employee ID: 2102
   Employee DOB: 19/01/1994
   Salary : 35000
..............................
PANKAJs-iMac:oops pankaj_kumar$
```

16.2

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
Enter the elements of 3*3 matrix:
m[0][0] = 1
m[0][1] = 2
m[0][2] = 3
m[1][0] = 4
m[1][1] = 5
m[1][2] = 6
m[2][0] = 7
m[2][1] = 8
m[2][2] = 9

The matrix is as follows:
1       2       3
4       5       6
7       8       9       Enter the elements of 3*3 matrix:
m[0][0] = 1
m[0][1] = 2
m[0][2] = 3
m[1][0] = 4
m[1][1] = 5
m[1][2] = 6
m[2][0] = 7
m[2][1] = 8
m[2][2] = 9
```

```
The matrix is as follows:
1       2       3
4       5       6
7       8       9
Transposed matrix:
1       4       7
2       5       8
3       6       9
Multiplication on matrixs:
30      36      42
66      81      96
102     126     150
PANKAJs-iMac:oops pankaj_kumar$
```

Q17.

Design a **Point** class with co-ordinates x and y representing point on plane (use constructors and overload << and >> operators). Design a **Polar** class which represent class using polar coordinates radius(r) and angle(a). Write program which takes input data for a Point object and perform **type conversion** so as to convert the Point data into Polar data.

Use the following formulas where required:

x = r * cos(a);
y = r * sin(a);
a = atan(y/x);
r = sqrt(x*x + y*y);

Sol-

```
#include <iostream>
#include <iomanip>
#include <cmath>

using namespace std;

class Point{
  float x;
  float y;
  float a;
  float r;
public:
  void convert(){
    a = atan(y/x);
    r = sqrt(x*x + y*y);
  }
  friend ostream & operator << (ostream &out, const Point &P);
  friend istream & operator >> (istream &in, Point &P);
};

ostream & operator << (ostream &out, const Point &P){
    out<<"  In degrees = "<<P.a<<endl;
    out<<"  In radians = "<<P.r<<endl;
    return out;
}

istream & operator >> (istream &in, Point &P){
    cout << "  Enter the value of x coordinate :";
    in >> P.x;
    cout << "  Enter the value of y coordinate :";
    in >> P.y;
    return in;
}

int main(){
  Point p;
  cin >> p;
  p.convert();
  cout << p;
  return 0;
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Enter the value of x coordinate :5
  Enter the value of y coordinate :5
  In degrees = 0.785398
  In radians = 7.07107
PANKAJs-iMac:oops pankaj_kumar$
```

Q18.

Write a base class with protected class members length and breadth. Derived class should have a function to calculate the area using length and breadth of base class

Sol-

```
#include <iostream>

using namespace std;

class Base{
protected:
	float length;
	float breadth;
	float area;
};

class Derived : protected Base{
public:
	friend ostream & operator << (ostream &out, const Derived &D);
	friend istream & operator >> (istream &in, Derived &D);
	void volume(){
		area = length*breadth;
	}
};

ostream & operator << (ostream &out, const Derived &B){
	out << "  Area is : "<<B.area<<endl;
	return out;
}

istream & operator >> (istream &in, Derived &B){
	cout<<"  Enter length : ";
	in >> B.length;
	cout<<"  Enter breadth : ";
	in >> B.breadth;
	return in;
}

int main(){
	Derived D;
	cin>>D;
	D.volume();
	cout<<D;
	return 0;
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Enter length : 6
  Enter breadth : 6
  Area is : 36
PANKAJs-iMac:oops pankaj_kumar$
```

Q19.

Write a base and derived class with simple integer class members. Implement concept of base class constructor initialization in the derived class constructor. Both the classes should have member function to display their class member values. This method should be protected in the base class and invoked from the derived class.

Sol-

```
#include <iostream>

using namespace std;

class Base{
protected:
	int x;
	int y;
	Base(int x1,int y1){
		x = x1;
		y = y1;
	}
	void display(){
		cout<<"  Base class"<<endl;
		cout<<"  X : "<<x<<" |  Y : "<<y<<endl;
	}
};

class Derived : private Base{
	int a;
	int b;
public:
	Derived (int x1,int y1,int a1,int b1) : Base(x1,y1){
		a = a1;
		b = b1;
	}
	void display( Derived &d){
		d.Base::display();
	}
	void display(){
		cout<<"  Derived class"<<endl;
		cout<<"  A : "<<a<<" |  B : "<<b<<endl;
	}
};

int main(){
	Derived D(1,2,3,4);
	D.display(D);
	D.display();
	return 0;
}
```

Q20.

Implement multiple and multi-level inheritance. Derived class **derived** should have protected and private inheritance from base classes base1 and base2, where base1 and base2 have protected data members. Derived class **derivedL2** should in turn publicly inherit from class **derived**. Try and access protected data members of base1 and base2 classes in **derivedL2**. Write destructor functions to see the sequence of destructor calls.

Sol-

```
#include <iostream>

using namespace std;

class base1{
protected:
	int x;
	base1(int x1){
		x = x1;
	}
	void display_1(){
		cout<<"  Base1 : "<<x<<endl;
	}
};

class base2{
protected:
	int y;
	base2(int x2){
```

```
                y = x2;
        }
        void display_2(){
                cout<<"  Base2 : "<<y<<endl;
        }
};

class derived : protected base1, private base2{
        int a;
public:
        derived(int x1, int x2, int x3) : base1(x1),base2(x2){
                a = x3;
        }
        void display_d(derived &d){
                d.base1::display_1();
                d.base2::display_2();
        }
        void display(){
                cout<<"  Derived : "<<a<<endl;
        }
};

class derivedL2 : public derived{
        int b;
public:
        derivedL2(int x1, int x2, int x3, int x4) : derived(x1,x2,x3){
                b = x4;
        }
        void display(derivedL2 &d){
                d.derived::display();
        }
        void display2(){
                cout<<"  DerivedL2 : "<<b<<endl;
        }
    ~derivedL2(){
        cout<<"  Delete derivedL2....."<<endl;
    }
};

int main(){
        derivedL2 d(1,2,3,4);
        d.display_d(d);
        d.display(d);
        d.display2();
        return 0;
}
```

OUTPUT

```
./PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Base1 : 1
  Base2 : 2
  Derived : 3
  DerivedL2 : 4
  Delete derivedL2.....
PANKAJs-iMac:oops pankaj_kumar$
```

Q21.

Defining class **Manager** which inherits from **Employee.** Manager has additional attributes **(bonus, employees_managed)** and behavior as (**getbonus, setbonus, addemployees_managed and getemployees_managed)**. Write the corresponding functions and test them. Write corresponding constructors, destructors and Operator overloading functions.

Sol-

```
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <cmath>

using namespace std;
```

```
int emp =0;

class Date{
	int dd,mm,yyyy;
public:
	void putdob(){
		cout<<"  Enter Date of Birth : "<<endl;
		cout<<"  Enter Date : ";
		cin>>dd;
		cout<<"  Enter Month : ";
		cin>>mm;
		cout<<"  Enter Year : ";
		cin>>yyyy;
	}

	void display_d(){
		cout<<"  Date of Birth : "<<dd<<"/"<<mm<<"/"<<yyyy<<endl;
	}
};

class Employee{
protected:
	char emp_name[30];
	int emp_id;
	float emp_salary;
	Date dob;
public:
/*	Employee(char *n, int i, int s, int d, int m, int y) : dob(d,m,y){
		int len = strlen(n);
		emp_name = new char [len + 1];
		strcpy(emp_name,n);
		emp_id = i;
		salary = s;
	}
*/
	void addemployees_detail(){
		cout<<"  Enter the name of Employee : ";
		cin>>emp_name;
		cout<<"  Enter ID of Employee : ";
		cin>>emp_id;
		cout<<"  Enter salary of Employee : ";
		cin>>emp_salary;
		dob.putdob();
	}

	void setsalary(int s){
		emp_salary = s;
	}

	int getsalary(){
		return emp_salary;
	}

	void display_emp_detail(){
		cout<<"  Name : "<<emp_name<<endl;
		cout<<"  ID : "<<emp_id<<endl;
		cout<<"  Salary : "<<emp_salary<<endl;
		dob.display_d();
	}
};

class Manager:protected Employee{
	float bonus;
	int no_emp;
	Employee e[5];
public:
	int m;
	int i;
	float salary;
	void getbonus();
	void setbonus(int bonus);
	void addmanager_detail(char * n){
		char c;
```

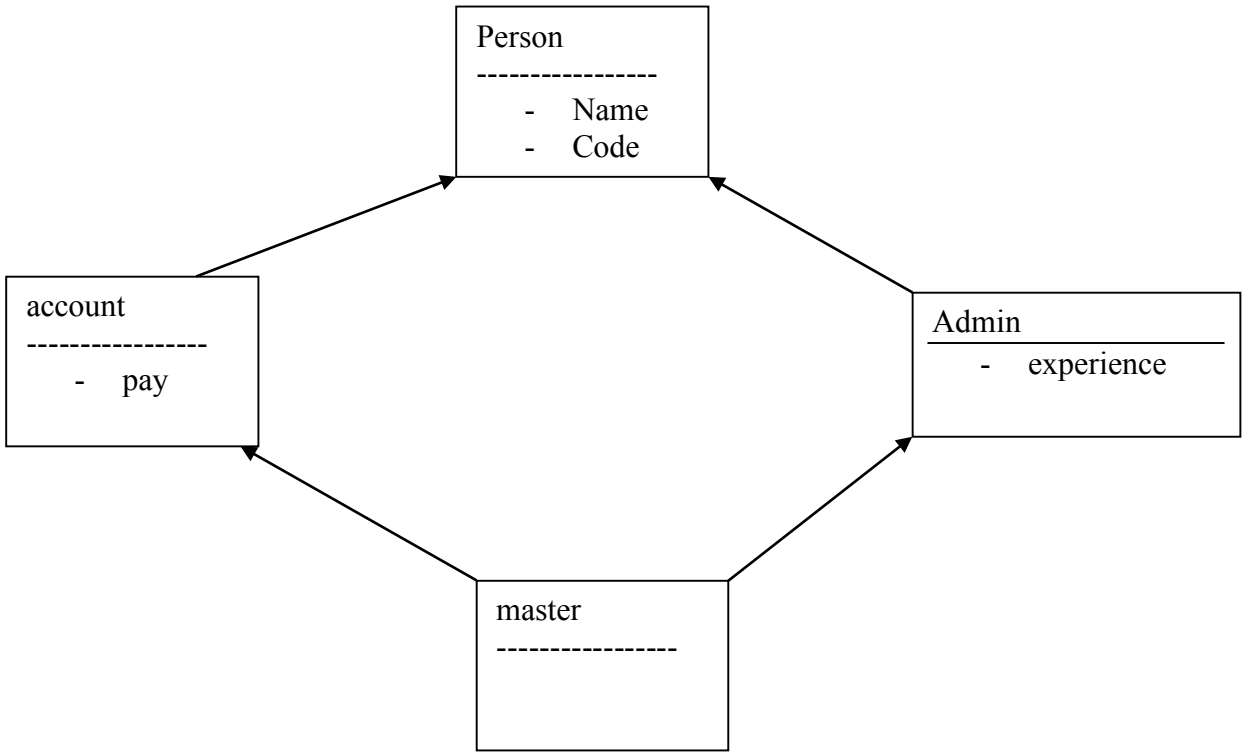
```
                strcpy(emp_name,n);
                cout<<"  Enter ID of Manager : ";
                cin>>emp_id;
                cout<<"  Enter salary of Manager : ";
                cin>>salary;
                setsalary(salary);
                cout<<"  Enter bonus : ";
                cin>>bonus;
                setbonus(bonus);
                dob.putdob();
                cout<<"  Do you want to add employee to the Manager (Y/N) : ";
                cin>>c;
                if(c=='Y'||c=='y'){
                        addemployees_managed();
                }
                else{
                        no_emp = emp;
                }
        }

        void addemployees_managed(){
                cout<<"  How many employee do you want to add to this Manager (Max. is 5)
: ";
                cin>>m;
                for(i=0; i<m; i++){
                        e[i].addemployees_detail();
                        no_emp = emp+1;
                        emp++;
                }
        }

        void getemployees_managed(){
                for(int j=0; j<i; j++){
                        e[j].display_emp_detail();
                }
        }

        void display_m(){
                cout<<"  Manager detail "<<endl<<endl;
                display_emp_detail();
                cout<<"  Total Employee managed by Manager is : "<<no_emp<<endl;
        }

        void display_e(){
                cout<<endl<<"  Detail of Employees "<<endl;
                for (int i=0; i<m; i++){
                        e[i].display_emp_detail();
                }
        }
};

void Manager::getbonus(){
        cout<<"  Bonus is : "<<bonus<<endl;
        cout<<"  Total salary is : "<<getsalary();
}

void Manager::setbonus(int bonus){
        float s,total_salary=0;
        s = getsalary();
        total_salary = s+bonus;
        setsalary(total_salary);
}

int main(){
        char n[30];
        Manager m;
        cout<<"  Enter name of Manager : ";
        cin>>n;
        m.addmanager_detail(n);
        m.display_m();
        m.display_e();

}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Enter name of Manager : Amit
  Enter ID of Manager : 1
  Enter salary of Manager : 25000
  Enter bonus : 5000
  Enter Date of Birth :
  Enter Date : 25
  Enter Month : 10
  Enter Year : 1994
  Do you want to add employee to the Manager (Y/N) : y
  How many employee do you want to add to this Manager (Max. is 5) : 2
  Enter the name of Employee : Gagan
  Enter ID of Employee : 2
  Enter salary of Employee : 25000
  Enter Date of Birth :
  Enter Date : 12
  Enter Month : 9
  Enter Year : 1995
  Enter the name of Employee : Deepak
  Enter ID of Employee : 3
  Enter salary of Employee : 25000
  Enter Date of Birth :
  Enter Date : 19
  Enter Month : 8
  Enter Year : 1993
  Manager detail

  Name : Amit
  ID : 1
  Salary : 30000
  Date of Birth : 25/10/1994
  Total Employee managed by Manager is : 2

  Detail of Employees
  Name : Gagan
  ID : 2
  Salary : 25000
  Date of Birth : 12/9/1995
  Name : Deepak
  ID : 3
  Salary : 25000
  Date of Birth : 19/8/1993
PANKAJs-iMac:oops pankaj_kumar$
```

Q22.

Consider a class network at drawn below. Define all the four classes and write a program to create, update and display the information contained in the **master** objects. Write constructors, destructors and Operator overloading functions where applicable.



Sol-

```
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <cmath>

using namespace std;

class person{
protected:
        char * name;
        int id;
public:
        person(const char *n, int i):id(i){
                int len = strlen(n);
                name = new char[len+1];
                strcpy(name,n);
        }

        ~person(){
                cout<<"  Person is deleted."<<endl;
```

```
        }

        void display_p(){
                cout<<"  Name : "<<name<<endl;
                cout<<"  ID : "<<id<<endl;
        }
};

class account:virtual protected person{
protected:
        int pay;
public:
        account(const char * n, int i, int p):person(n,i),pay(p){};
        ~account(){
                cout<<"  Account is deleted."<<endl;
        }
        void display_a(){
                display_p();
                cout<<"  Pay : "<<pay<<endl;
        }
};

class admin:virtual protected person{
protected:
        int experience;
public:
        admin(const char * n, int i, int e):person(n,i){
                experience = e;
        }

        ~admin(){
                cout<<"  Admin is deleted."<<endl;
        }

        void display_ad(){
                display_p();
                cout<<"  Experience : "<<experience<<endl;
        }
};

class master:protected account, protected admin{

public:
        master(const char * n, int i, int p, int
e):account(n,i,p),admin(n,i,e),person(n,i){};
        ~master(){
                cout<<"  Master is deleted."<<endl;
        }
        void display(){
                cout<<"  Name : "<<name<<endl;
                cout<<"  ID : "<<id<<endl;
                cout<<"  Pay : "<<pay<<endl;
                cout<<"  Experience : "<<experience<<endl;
        }
};
int main(){
        master m("Pankaj",144,10,30000);
        m.display();
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Name : Pankaj
  ID : 144
  Pay : 10
  Experience : 30000
  Master is deleted.
  Admin is deleted.
  Account is deleted.
  Person is deleted.
PANKAJs-iMac:oops pankaj_kumar$
```

Q23.
Define a **Employee** class with class members as ID, name and salary. Define a member function "greater" which takes one parameter (object of class Employee) as function argument. The function should compare the salary of two Employees and return the object whose salary is greater. Achieve the above using **"this"** pointer.

Sol-

```
#include <iostream>
#include <cstring>

using namespace std;

class Employee{
	int emp_ID;
	char emp_name[30];
	float emp_salary;
public:
	Employee(int id, char *name, float s){
		emp_ID = id;
		strcpy(emp_name,name);
		emp_salary = s;
	}

	Employee & greater(Employee & E){
		if(E.emp_salary >= emp_salary)
			return E;
		else
			return *this;
	}

	void display(){
		cout<<"  Employee ID : "<<emp_ID<<endl;
		cout<<"  Employee name : "<<emp_name<<endl;
		cout<<"  Employee Salary : "<<emp_salary<<endl;
	}

};

int main(){
	Employee E1(1,"Pankaj",30000),E2(2,"Gagan",35000);
	E1.display();
	E2.display();
	cout<<endl<<"*********************************"<<endl<<endl;
	Employee E3 = E1.greater(E2);
	cout<<"  Greater salary has "<<endl;
	E3.display();
	cout<<endl;
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Employee ID : 1
  Employee name : Pankaj
  Employee Salary : 30000
  Employee ID : 2
  Employee name : Gagan
  Employee Salary : 35000

*********************************

  Greater salary has
  Employee ID : 2
  Employee name : Gagan
  Employee Salary : 35000

PANKAJs-iMac:oops pankaj_kumar$
```

Q24.
Define a class shape with 2 data members to store length and height. Define member functions get_data() and display_area(). Display_area() should be virtual function. Define 2 derived classes from shape, triangle and rectangle which redefine display_area() for their own specific area calculation.

Sol-

```
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <cmath>

using namespace std;

class shape{
	float length;
	float height;
	float f_area;
public:
	virtual void get_data(){
		cout<<"*******************************"<<endl;
		cout<<"  Enter length : ";
		cin>>length;
		cout<<"  Enter height : ";
		cin>>height;
	}

	virtual void display_area(){
		cout<<f_area<<endl;
	}

	float ar(){
		return (length*height);
	}

	virtual void area(float a){
		f_area = a;
	}
};

class triangle:public shape{

public:
	void getdata(){
		shape::get_data();
	}

	void area(){
		shape::area((0.5)*ar());
	}

	void display_area(){
		cout<<"  Area of Triangle : ";
		shape::display_area();
		cout<<"*******************************"<<endl;
	}
};

class rectangle:public shape{

public:
	void getdata(){
		shape::get_data();
	}

	void area(){
		shape::area(ar());
	}

	void display_area(){
		cout<<"  Area of Rectangle : ";
```

```
                shape::display_area();
                cout<<"*******************************"<<endl;
        }

};

int main(){
        int ch;
        triangle t;
        rectangle r;
        do{
                cout<<endl<<"  1.Calculate area of Triangle."<<endl;
                cout<<"  2.Calculate area of Rectangle."<<endl;
                cout<<"  3.Exit."<<endl<<endl;
                cout<<"  Enter your choice : ";
                cin>>ch;
                switch(ch){
                        case 1 :
                        t.getdata();
                        t.area();
                        t.display_area();
                        break;

                        case 2 :
                        r.getdata();
                        r.area();
                        r.display_area();
                        break;

                        case 3 :
                        exit(0);

                        default:
                        cout<<"  Enter correct choice."<<endl<<endl;
                }
        }while(true);
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out

  1.Calculate area of Triangle.
  2.Calculate area of Rectangle.
  3.Exit.

  Enter your choice : 1
*******************************
  Enter length : 18
  Enter height : 9
  Area of Triangle : 81
*******************************

  1.Calculate area of Triangle.
  2.Calculate area of Rectangle.
  3.Exit.

  Enter your choice : 2
*******************************
  Enter length : 13
  Enter height : 13
  Area of Rectangle : 169
*******************************

  1.Calculate area of Triangle.
  2.Calculate area of Rectangle.
  3.Exit.

  Enter your choice : 3
PANKAJs-iMac:oops pankaj_kumar$
```

Q25.
Extend the above program to calculate area of a circle by introducing a new derived class called circle. The get_data() function in the shape class can be modified by taking default value for the second parameter.

Sol-

```
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <cmath>

using namespace std;

class shape{
	float length;
	float height;
	float f_area;
public:
	virtual void get_data(){
		cout<<"*******************************"<<endl;
		cout<<"  Enter length : ";
		cin>>length;
		cout<<"  Enter height : ";
		cin>>height;
	}

	virtual void get_data_c(){
		cout<<"*******************************"<<endl;
		cout<<"  Enter radius : ";
		cin>>length;
		height = length;
	}

	virtual void display_area(){
		cout<<f_area<<endl;
	}

	virtual float ar(){
		return (length*height);
	}

	virtual void area(float a){
		f_area = a;
	}
};

class triangle:public shape{

public:
	void get_data(){
		shape::get_data();
	}

	void area(){
		shape::area((0.5)*ar());
	}

	void display_area(){
		cout<<"  Area of Triangle : ";
		shape::display_area();
		cout<<"*******************************"<<endl;
	}
};

class rectangle:public shape{

public:
	void get_data(){
		shape::get_data();
	}

	void area(){
```

```
                shape::area(ar());
        }

        void display_area(){
                cout<<"  Area of Rectangle : ";
                shape::display_area();
                cout<<"*******************************"<<endl;
        }

};

class circle:public shape{

public:
        void get_data(){
                shape::get_data_c();
        }

        void area(){
                shape::area(3.14159265359*ar());
        }

        void display_area(){
                cout<<"  Area of Circle : ";
                shape::display_area();
                cout<<"*******************************"<<endl;
        }

};

int main(){
        int ch;
        triangle t;
        rectangle r;
        circle c;
        do{
                cout<<endl<<"  1.Calculate area of Triangle."<<endl;
                cout<<"  2.Calculate area of Rectangle."<<endl;
                cout<<"  3.Calculate area of Circle."<<endl;
                cout<<"  4.Exit."<<endl<<endl;
                cout<<"  Enter your choice : ";
                cin>>ch;
                switch(ch){
                        case 1 :
                        t.get_data();
                        t.area();
                        t.display_area();
                        break;

                        case 2 :
                        r.get_data();
                        r.area();
                        r.display_area();
                        break;

                        case 3 :
                        c.get_data();
                        c.area();
                        c.display_area();
                        break;

                        case 4 :
                        exit(0);

                        default:
                        cout<<"  Enter correct choice."<<endl<<endl;
                }
        }while(true);
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out

  1.Calculate area of Triangle.
  2.Calculate area of Rectangle.
  3.Calculate area of Circle.
  4.Exit.

  Enter your choice : 1
*******************************
  Enter length : 18
  Enter height : 5
  Area of Triangle : 45
*******************************

  1.Calculate area of Triangle.
  2.Calculate area of Rectangle.
  3.Calculate area of Circle.
  4.Exit.

  Enter your choice : 2
*******************************
  Enter length : 6
  Enter height : 16
  Area of Rectangle : 96
*******************************

  1.Calculate area of Triangle.
  2.Calculate area of Rectangle.
  3.Calculate area of Circle.
  4.Exit.

  Enter your choice : 3
*******************************
  Enter radius : 5
  Area of Circle : 78.5398
*******************************

  1.Calculate area of Triangle.
  2.Calculate area of Rectangle.
  3.Calculate area of Circle.
  4.Exit.

  Enter your choice : 4
PANKAJs-iMac:oops pankaj_kumar$
```

Q7.5.

Sol-

```
#include <iostream>
#include <iomanip>
#include <cmath>
#include <cstdlib>

using namespace std;

class Polar;

class Rectangle{
  double x;
  double y;
public:
       friend Rectangle convert_p_to_r( Polar &P);
       friend Polar convert_r_to_p( Rectangle &R);
       friend ostream & operator << (ostream &out, const Rectangle &P);
       friend istream & operator >> (istream &in, Rectangle &P);
};

class Polar{
       double a;
       double r;
```

```
public:
       friend Rectangle convert_p_to_r( Polar &P);
       friend Polar convert_r_to_p( Rectangle &R);
       friend ostream & operator << (ostream &out, const Polar &P);
  friend istream & operator >> (istream &in, Polar &P);
};

Rectangle convert_p_to_r( Polar &P ){
  Rectangle R;
  R.x = P.r * cos (P.a);
  R.y = P.r * sin (P.a);
  return R;
}

Polar convert_r_to_p( Rectangle &R ){
       Polar P;
       P.a = atan(R.y/R.x);
  P.r = sqrt(R.x*R.x + R.y*R.y);
  return P;
}

ostream & operator << (ostream &out, const Rectangle &R){
    out<<"  Your answer in Radians is : "<<endl;
    out<<"  x coordinate = "<<R.x<<endl;
    out<<"  y coordinate = "<<R.y<<endl;
    return out;
}

istream & operator >> (istream &in, Polar &P){
    cout << "  Enter the value in degrees :";
    in >> P.a;
    cout << "  Enter the value in radians :";
    in >> P.r;
    return in;
}

ostream & operator << (ostream &out, const Polar &P){
    out<<"  Your answer in Polar is : "<<endl;
    out<<"  In degrees = "<<P.a<<endl;
    out<<"  In radians = "<<P.r<<endl;
    return out;
}

istream & operator >> (istream &in, Rectangle &P){
    cout << "  Enter the value of x coordinate :";
    in >> P.x;
    cout << "  Enter the value of y coordinate :";
    in >> P.y;
    return in;
}

int main(){
  int c;
  do{
    cout<<endl<<"  1.Convert rectangle to polar."<<endl;
    cout<<"  2.Convert polar to rectangle."<<endl;
    cout<<"  3.Exit."<<endl<<endl;
    cout<<"  Enter your choice: ";
    cin>>c;
    switch(c){
        case 1:
         Rectangle R;
         cin>>R;
         Polar P1;
         P1=convert_r_to_p(R);
         cout<<P1;
         break;

      case 2:
         Polar P;
         cin>>P;
         Rectangle R1;
         R1=convert_p_to_r(P);
         cout<<R1;
```

```
        break;

      case 3:
      exit(0);
      break;

      default :
      cout<<"  You enter a worng choice. Please try again later or select correct
choice."<<endl;
    }
  }while(true);
}
```

Q8.1.

Sol-

```
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <cmath>

using namespace std;
int account_no_c = 13100;
int account_no_s = 15100;
int cheque_n = 274010;

class account{
       char *c_name;
       char *type_acc;
public:
       void putdetail(char *name, char *type){
              int len_n = strlen(name);
              int len_t = strlen(type);
              c_name = new char[len_n + 1];
              type_acc = new char[len_t + 1];
              strcpy(c_name, name);
              strcpy(type_acc,type);
       }

       void display(){
              cout<<"  Account holder name : "<<c_name<<endl;
              cout<<"  Account Type : "<< type_acc<<endl;
       }
};

class cur_acct:public account{
       int cheque_no_f;
       int cheque_no_t;
       int acc_no_c;
       float balance_c;
public:
       void putdetail_c();
       int cheque_no();
       void deposit_amount();
       void issue();
       void withd_cheque(int cheque);
       void withd_cash();
       void check_bal(float min_bal);
       void display_bala_c();
       int check_acc_no(){
       return acc_no_c;
       }
       void display_c();
};

class sav_acct:public account{
       float ci;
       int acc_no_s;
       float balance_s;
       char *withd_status_s;
public:
```

```
        void putdetail_s();
        void deposit_amount();
        void calc_ci();
        void withdraw();
        void display_bala_s();
        int check_acc_no(){
        return acc_no_s;
        }
        void display_s();
};

int cur_acct::cheque_no(){
        return cheque_no_f;
}

void cur_acct::display_bala_c(){
        cout<<endl<<"*******************************************"<<endl;
        display();
        cout<<"  Total Balance is : "<<balance_c<<endl<<endl;
        cout<<endl<<"*******************************************"<<endl;
}

void cur_acct::putdetail_c(){
        char name[30],type[10];
        cout<<"  Enter the name of account holder : ";
        cin.ignore(-1);
        cin.getline(name,30);
        cout<<"  Enter account type : ";
        cin.ignore(-1);
        cin.getline(type,10);
        putdetail(name,type);
        acc_no_c = account_no_c;
        cheque_no_f = cheque_n-10;
        cheque_no_t = cheque_n;
        balance_c = 5000;
        account_no_c++;
        cheque_n+=100;
}

void cur_acct::display_c(){
        cout<<"*******************************************"<<endl;
        display();
        cout<<"  Account number : "<<acc_no_c<<endl;
        cout<<"  Total Balance : "<<balance_c<<endl;
        if(cheque_no_f == cheque_no_t){
                cout<<"  Cheque no. : "<<cheque_no_f<<" is cheque so please issue a new
cheque book."<<endl;
        }
        else
          {
        cout<<"  Cheque Book no. From : "<<cheque_no_f<<" to "<<cheque_no_t<<endl;
        }
        cout<<"*******************************************"<<endl;
}

void cur_acct::check_bal(float min_bal){
        if(min_bal<5000){
                cout<<"  Your account has gone below minimum balance required. That is
5000."<<endl<<endl;
                balance_c -= 100;
        }
}

void cur_acct::issue(){
        cheque_no_f -= 10;
}

void cur_acct::withd_cheque(int cheque){
        float amount;
                cout<<"   Enter the amount to withdraw from acccount : ";
                cin>>amount;
                balance_c-=amount;
                check_bal(balance_c);
                if(cheque == cheque_no_t){
```

```
                        cout<<"  This was your last cheque, So please issue new cheque
book."<<endl;
                        issue();
                }
                else
                {
                        cheque_no_f++;
                }
                display_c();
}

void cur_acct::deposit_amount(){
        double amount;
        cout<<"Enter amount to deposit in account :";
        cin>>amount;
        balance_c+=amount;
        display_c();
}

void cur_acct::withd_cash(){
        float amount;
        cout<<"   Enter the amount to withdraw from acccount : ";
        cin>>amount;
        balance_c-=amount;
        check_bal(balance_c);
        display_c();
}

void sav_acct::display_bala_s(){
        cout<<endl<<"*******************************************"<<endl;
        display();
        cout<<"  Total Balance is : "<<balance_s<<endl<<endl;
        cout<<endl<<"*******************************************"<<endl;
}

void sav_acct::putdetail_s(){
        char name[30],type[10];
        cout<<"  Enter the name of account holder : ";
        cin.ignore(-1);
        cin.getline(name,30);
        cout<<"  Enter accoutn type : ";
        cin.ignore(-1);
        cin.getline(type,10);
        putdetail(name,type);
        acc_no_s = account_no_s;
        balance_s = 2000;
        account_no_s++;
        ci = 0.0;
}

void sav_acct::display_s(){
        cout<<"*******************************************"<<endl;
        display();
        cout<<"  Account number : "<<acc_no_s<<endl;
        cout<<"  Total Balance : "<<balance_s<<endl;
        cout<<"*******************************************"<<endl;
}

void sav_acct::calc_ci(){
        float time;
        cout<<"   Enter time : ";
        cin>>time;
        ci = balance_s * pow((1+0.05),time) - balance_s;
        display_s();
        balance_s+=ci;
        cout<<"   Compound Interest is : "<<ci<<endl;
        cout<<"   Main balance is : "<<balance_s<<endl<<endl<<endl;
}

void sav_acct::deposit_amount(){
        double amount;
        cout<<"Enter amount to deposit in account :";
        cin>>amount;
        balance_s+=amount;
```

```
	display_s();
}

void sav_acct::withdraw(){
	float amount;
	cout<<"   Enter the amount to withdraw from account : ";
	cin>>amount;
	balance_s -= amount;
	display_s();
}

int main(){
	int type,c_c;
	int account_no;
	float amount;
	int c=2,s=1,ch,j;
	cur_acct c_a[c];
	sav_acct s_a[s];
	cout<<"  Enter detail of two current account holder and one saving account holder
first."<<endl;
	for(int i=0; i<c; i++){
	c_a[i].putdetail_c();
	}
	for(int j=0; j<s; j++){
	s_a[j].putdetail_s();
	}
	do{
		cout<<endl<<endl<<"  1.Deposit money in account."<<endl;
		cout<<"  2.Display the balance."<<endl;
		cout<<"  3.Compute and deposit Interest in Saving account."<<endl;
		cout<<"  4.Withdraw money from account."<<endl;
		cout<<"  5.Display detail of account holder."<<endl;
		cout<<"  6.Exit."<<endl<<endl;
		cout<<"    Enter your choice : ";
		cin>>ch;
		switch(ch){
			case 1 :
			cout<<endl<<"  Enter account type 1 for current and 2 for savings :
";
			cin>>type;
			if(type==1){
				cout<<endl<<"  Please enter your account no. : ";
				cin>>account_no;
				for(j=0; j<c; j++){
					if(c_a[j].check_acc_no()==account_no)
					break;
				}
				c_a[j].deposit_amount();
				}
				else
					if(type==2){
						cout<<endl<<"  Please enter your account no.
: ";
						cin>>account_no;
						for(j=0; j<s; j++){
							if(s_a[j].check_acc_no()==account_no)
							break;
						}
						s_a[j].deposit_amount();
					}
					else
					{
						cout<<"  Please enter correct account type.
Please try again later......"<<endl;
					}
					break;

			case 2 :
			cout<<endl<<"  Enter account type 1 for current and 2 for savings :
";
			cin>>type;
			if(type==1){
				cout<<endl<<"  Please enter your account no. : ";
				cin>>account_no;
```

```
                        for(j=0; j<c; j++){
                                if(c_a[j].check_acc_no()==account_no)
                                break;
                        }
                        c_a[j].display_bala_c();
                        }
                        else
                                if(type==2){
                                        cout<<endl<<"  Please enter your account no.
: ";

                                        cin>>account_no;
                                        for(j=0; j<s; j++){
                                                if(s_a[j].check_acc_no()==account_no)
                                                break;
                                        }
                                        s_a[j].display_bala_s();
                                }
                                else
                                {
                                        cout<<"  Please enter correct account type.
Please try again later......"<<endl;
                                }
                                break;

                case 3 :
                cout<<endl<<"  Please enter your account no. : ";
                cin>>account_no;
                for(j=0; j<c; j++){
                        if(s_a[j].check_acc_no()==account_no)
                        break;
                        }
                        s_a[j].calc_ci();
                        break;

                case 4 :
                cout<<endl<<"  Enter account type 1 for current and 2 for savings :
";
                cin>>type;
                if(type==1){
                        cout<<"  1.By Cheque."<<endl;
                        cout<<"  2.By Self."<<endl;
                        cout<<"   Enter your choice : ";
                        cin>>c_c;
                        switch(c_c){
                                case 1 :
                                cout<<endl<<"  Please enter your account no. : ";
                                cin>>account_no;
                                for(j=0; j<c; j++){
                                if(c_a[j].check_acc_no()==account_no)
                                break;
                                }
                            c_a[j].withd_cheque(c_a[j].cheque_no());
                            break;

                                case 2 :
                                cout<<endl<<"  Please enter your account no. : ";
                                cin>>account_no;
                                        for(j=0; j<c; j++){
                                if(c_a[j].check_acc_no()==account_no)
                                break;
                                }
                                c_a[j].withd_cash();
                                break;
                        }
                }
                        else
                                if(type==2){
                                        cout<<endl<<"  Please enter your account no.
: ";

                                        cin>>account_no;
                                        for(j=0; j<s; j++){
                                                if(s_a[j].check_acc_no()==account_no)
                                                break;
                                        }
```

```
                                        s_a[j].withdraw();
                                }
                                else
                                {
                                        cout<<"  Please enter correct account type.
Please try again later......"<<endl;
                                }
                                break;

                        case 5 :
                        for(int i=0; i<c; i++){
                                c_a[i].display_c();
                        }
                        for(int i=0; i<s; i++){
                                s_a[i].display_s();
                        }
                        break;

                        case 6 :
                        exit(0);
                        }
                }while(true);

}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Enter detail of two current account holder and one saving account holder first.
  Enter the name of account holder : Pankaj
  Enter account type : current
  Enter the name of account holder : Amit
  Enter account type : current
  Enter the name of account holder : Gagan
  Enter accoutn type : saving


  1.Deposit money in account.
  2.Display the balance.
  3.Compute and deposit Interest in Saving account.
  4.Withdraw money from account.
  5.Display detail of account holder.
  6.Exit.

    Enter your choice : 1

  Enter account type 1 for current and 2 for savings : 1

  Please enter your account no. : 13100
Enter amount to deposit in account :5000
****************************************
  Account holder name : Pankaj
  Account Type : current
  Account number : 13100
  Total Balance : 10000
  Cheque Book no. From : 274000 to 274010
****************************************


  1.Deposit money in account.
  2.Display the balance.
  3.Compute and deposit Interest in Saving account.
  4.Withdraw money from account.
  5.Display detail of account holder.
  6.Exit.

    Enter your choice : 5
****************************************
  Account holder name : Pankaj
  Account Type : current
  Account number : 13100
  Total Balance : 10000
  Cheque Book no. From : 274000 to 274010
****************************************
```

```
****************************************
  Account holder name : Amit
  Account Type : current
  Account number : 13101
  Total Balance : 5000
  Cheque Book no. From : 274100 to 274110
****************************************
****************************************
  Account holder name : Gagan
  Account Type : saving
  Account number : 15100
  Total Balance : 2000
****************************************


  1.Deposit money in account.
  2.Display the balance.
  3.Compute and deposit Interest in Saving account.
  4.Withdraw money from account.
  5.Display detail of account holder.
  6.Exit.

    Enter your choice : 4

  Enter account type 1 for current and 2 for savings : 2

  Please enter your account no. : 15100
   Enter the amount to withdraw from account : 1000
****************************************
  Account holder name : Gagan
  Account Type : saving
  Account number : 15100
  Total Balance : 1000
****************************************


  1.Deposit money in account.
  2.Display the balance.
  3.Compute and deposit Interest in Saving account.
  4.Withdraw money from account.
  5.Display detail of account holder.
  6.Exit.

    Enter your choice : 3

  Please enter your account no. : 15100
   Enter time : 2
****************************************
  Account holder name : Gagan
  Account Type : saving
  Account number : 15100
  Total Balance : 1000
****************************************
   Compound Interest is : 102.5
   Main balance is : 1102.5




  1.Deposit money in account.
  2.Display the balance.
  3.Compute and deposit Interest in Saving account.
  4.Withdraw money from account.
  5.Display detail of account holder.
  6.Exit.

    Enter your choice : 6
PANKAJs-iMac:oops pankaj_kumar$
```

Q8.2.

Sol-

```
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <cmath>

using namespace std;
int account_no_c = 13100;
int account_no_s = 15100;
int cheque_n = 274010;

class account{
        char *c_name;
        char *type_acc;
public:
        account(char *name, char *type){
                int len_n = strlen(name);
                int len_t = strlen(type);
                c_name = new char[len_n + 1];
                type_acc = new char[len_t + 1];
                strcpy(c_name, name);
                strcpy(type_acc,type);
        }
        void putdetail(char *name, char *type){
                int len_n = strlen(name);
                int len_t = strlen(type);
                c_name = new char[len_n + 1];
                type_acc = new char[len_t + 1];
                strcpy(c_name, name);
                strcpy(type_acc,type);
        }

        void display(){
                cout<<"  Account holder name : "<<c_name<<endl;
                cout<<"  Account Type : "<< type_acc<<endl;
        }
};

class cur_acct:public account{
        int cheque_no_f;
        int cheque_no_t;
        int acc_no_c;
        float balance_c;
public:
        cur_acct(char *name, char *type):account(name,type){
                acc_no_c = account_no_c;
                cheque_no_f = cheque_n-10;
                cheque_no_t = cheque_n;
                balance_c = 5000;
                account_no_c++;
                cheque_n+=100;
        }

        void putdetail_c();
        int cheque_no();
        void deposit_amount();
        void issue();
        void withd_cheque(int cheque);
        void withd_cash();
        void check_bal(float min_bal);
        void display_bala_c();
        int check_acc_no(){
        return acc_no_c;
        }
        void display_c();
};

class sav_acct:public account{
        float ci;
        int acc_no_s;
        float balance_s;
```

```
        char *withd_status_s;
public:
        sav_acct(char *name, char *type):account(name,type){
                acc_no_s = account_no_s;
                balance_s = 2000;
                account_no_s++;
        }
        void putdetail_s();
        void deposit_amount();
        void calc_ci();
        void withdraw();
        void display_bala_s();
        int check_acc_no(){
        return acc_no_s;
        }
        void display_s();
};

int cur_acct::cheque_no(){
        return cheque_no_f;
}

void cur_acct::display_bala_c(){
        cout<<endl<<"*********************************************"<<endl;
        display();
        cout<<"  Total Balance is : "<<balance_c<<endl<<endl;
        cout<<endl<<"*********************************************"<<endl;
}

void cur_acct::putdetail_c(){
        char name[30],type[10];
        cout<<"  Enter the name of account holder : ";
        cin.ignore(-1);
        cin.getline(name,30);
        cout<<"  Enter account type : ";
        cin.ignore(-1);
        cin.getline(type,10);
        putdetail(name,type);
        acc_no_c = account_no_c;
        cheque_no_f = cheque_n-10;
        cheque_no_t = cheque_n;
        balance_c = 5000;
        account_no_c++;
        cheque_n+=100;
}

void cur_acct::display_c(){
        cout<<"*********************************************"<<endl;
        display();
        cout<<"  Account number : "<<acc_no_c<<endl;
        cout<<"  Total Balance : "<<balance_c<<endl;
        if(cheque_no_f == cheque_no_t){
                cout<<"  Cheque no. : "<<cheque_no_f<<" is cheque so please issue a new
cheque book."<<endl;
        }
        else
          {
        cout<<"  Cheque Book no. From : "<<cheque_no_f<<" to "<<cheque_no_t<<endl;
        }
        cout<<"*********************************************"<<endl;
}

void cur_acct::check_bal(float min_bal){
        if(min_bal<5000){
                cout<<"  Your account has gone below minimum balance required. That is
5000."<<endl<<endl;
                balance_c -= 100;
        }
}

void cur_acct::issue(){
        cheque_no_f -= 10;
}
```

```
void cur_acct::withd_cheque(int cheque){
	float amount;
		cout<<"   Enter the amount to withdraw from acccount : ";
		cin>>amount;
		balance_c-=amount;
		check_bal(balance_c);
		if(cheque == cheque_no_t){
			cout<<"  This was your last cheque, So please issue new cheque
book."<<endl;
			issue();
		}
		else
		{
			cheque_no_f++;
		}
		display_c();
}

void cur_acct::deposit_amount(){
	double amount;
	cout<<"Enter amount to deposit in account :";
	cin>>amount;
	balance_c+=amount;
	display_c();
}

void cur_acct::withd_cash(){
	float amount;
	cout<<"   Enter the amount to withdraw from acccount : ";
	cin>>amount;
	balance_c-=amount;
	check_bal(balance_c);
	display_c();
}

void sav_acct::display_bala_s(){
	cout<<endl<<"*******************************************"<<endl;
	display();
	cout<<"  Total Balance is : "<<balance_s<<endl<<endl;
	cout<<endl<<"*******************************************"<<endl;
}

void sav_acct::putdetail_s(){
	char name[30],type[10];
	cout<<"  Enter the name of account holder : ";
	cin.ignore(-1);
	cin.getline(name,30);
	cout<<"  Enter accoutn type : ";
	cin.ignore(-1);
	cin.getline(type,10);
	putdetail(name,type);
	acc_no_s = account_no_s;
	balance_s = 2000;
	account_no_s++;
	ci = 0.0;
}

void sav_acct::display_s(){
	cout<<"*******************************************"<<endl;
	display();
	cout<<"  Account number : "<<acc_no_s<<endl;
	cout<<"  Total Balance : "<<balance_s<<endl;
	cout<<"*******************************************"<<endl;
}

void sav_acct::calc_ci(){
	float time;
	cout<<"   Enter time : ";
	cin>>time;
	ci = balance_s * pow((1+0.05),time) - balance_s;
	display_s();
	balance_s+=ci;
	cout<<"   Compound Interest is : "<<ci<<endl;
	cout<<"   Main balance is : "<<balance_s<<endl<<endl<<endl;
```

```
}

void sav_acct::deposit_amount(){
	double amount;
	cout<<"Enter amount to deposit in account :";
	cin>>amount;
	balance_s+=amount;
	display_s();
}

void sav_acct::withdraw(){
	float amount;
	cout<<"   Enter the amount to withdraw from account : ";
	cin>>amount;
	balance_s -= amount;
	display_s();
}

int main(){
	int type,c_c;
	int account_no;
	float amount;
	int c=2,s=1,ch,j;
	cur_acct c_a("Amit","current");
	sav_acct s_a("Pankaj","saving");
	do{
		cout<<endl<<endl<<"  1.Deposit money in account."<<endl;
		cout<<"  2.Display the balance."<<endl;
		cout<<"  3.Compute and deposit Interest in Saving account."<<endl;
		cout<<"  4.Withdraw money from account."<<endl;
		cout<<"  5.Display detail of account holder."<<endl;
		cout<<"  6.Exit."<<endl<<endl;
		cout<<"    Enter your choice : ";
		cin>>ch;
		switch(ch){
			case 1 :
			cout<<endl<<"  Enter account type 1 for current and 2 for savings :
";
			cin>>type;
			if(type==1){
				cout<<endl<<"  Please enter your account no. : ";
				cin>>account_no;
				for(j=0; j<1; j++){
					if(c_a.check_acc_no()==account_no)
					break;
				}
				c_a.deposit_amount();
				}
				else
					if(type==2){
						cout<<endl<<"  Please enter your account no.
: ";
						cin>>account_no;
						for(j=0; j<1; j++){
							if(s_a.check_acc_no()==account_no)
							break;
						}
						s_a.deposit_amount();
					}
					else
					{
						cout<<"  Please enter correct account type.
Please try again later......"<<endl;
					}
					break;

			case 2 :
			cout<<endl<<"  Enter account type 1 for current and 2 for savings :
";
			cin>>type;
			if(type==1){
				cout<<endl<<"  Please enter your account no. : ";
				cin>>account_no;
				for(j=0; j<1; j++){
```

```
                                if(c_a.check_acc_no()==account_no)
                                break;
                        }
                        c_a.display_bala_c();
                        }
                        else
                                if(type==2){
                                        cout<<endl<<"  Please enter your account no.
: ";
                                        cin>>account_no;
                                        for(j=0; j<1; j++){
                                                if(s_a.check_acc_no()==account_no)
                                                break;
                                        }
                                        s_a.display_bala_s();
                                }
                                else
                                {
                                        cout<<"  Please enter correct account type.
Please try again later......"<<endl;
                                }
                                break;

                        case 3 :
                        cout<<endl<<"  Please enter your account no. : ";
                        cin>>account_no;
                        for(j=0; j<1; j++){
                                if(s_a.check_acc_no()==account_no)
                                break;
                                }
                                s_a.calc_ci();
                                break;

                        case 4 :
                        cout<<endl<<"  Enter account type 1 for current and 2 for savings :
";
                        cin>>type;
                        if(type==1){
                                cout<<"  1.By Cheque."<<endl;
                                cout<<"  2.By Self."<<endl;
                                cout<<"   Enter your choice : ";
                                cin>>c_c;
                                switch(c_c){
                                        case 1 :
                                        cout<<endl<<"  Please enter your account no. : ";
                                        cin>>account_no;
                                        for(j=0; j<1; j++){
                                        if(c_a.check_acc_no()==account_no)
                                        break;
                                        }
                                    c_a.withd_cheque(c_a.cheque_no());
                                    break;

                                        case 2 :
                                        cout<<endl<<"  Please enter your account no. : ";
                                        cin>>account_no;
                                                for(j=0; j<1; j++){
                                        if(c_a.check_acc_no()==account_no)
                                        break;
                                        }
                                        c_a.withd_cash();
                                        break;
                                }
                        }
                                else
                                        if(type==2){
                                                cout<<endl<<"  Please enter your account no.
: ";
                                                cin>>account_no;
                                                for(j=0; j<1; j++){
                                                        if(s_a.check_acc_no()==account_no)
                                                        break;
                                                }
                                                s_a.withdraw();
```

```
                                    }
                                    else
                                    {
                                          cout<<"  Please enter correct account type.
Please try again later......"<<endl;
                                    }
                                    break;

                        case 5 :
                        for(int i=0; i<1; i++){
                              c_a.display_c();
                        }
                        for(int i=0; i<1; i++){
                              s_a.display_s();
                        }
                        break;

                        case 6 :
                        exit(0);
                        }
                }while(true);

}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out


  1.Deposit money in account.
  2.Display the balance.
  3.Compute and deposit Interest in Saving account.
  4.Withdraw money from account.
  5.Display detail of account holder.
  6.Exit.

    Enter your choice : 5
*****************************************
  Account holder name : Amit
  Account Type : current
  Account number : 13100
  Total Balance : 5000
  Cheque Book no. From : 274000 to 274010
*****************************************
*****************************************
  Account holder name : Pankaj
  Account Type : saving
  Account number : 15100
  Total Balance : 2000
*****************************************


  1.Deposit money in account.
  2.Display the balance.
  3.Compute and deposit Interest in Saving account.
  4.Withdraw money from account.
  5.Display detail of account holder.
  6.Exit.

    Enter your choice : 1

  Enter account type 1 for current and 2 for savings : 1

  Please enter your account no. : 13100
Enter amount to deposit in account :5000
*****************************************
  Account holder name : Amit
  Account Type : current
  Account number : 13100
  Total Balance : 10000
  Cheque Book no. From : 274000 to 274010
*****************************************
```

```
  1.Deposit money in account.
  2.Display the balance.
  3.Compute and deposit Interest in Saving account.
  4.Withdraw money from account.
  5.Display detail of account holder.
  6.Exit.

    Enter your choice : 1

  Enter account type 1 for current and 2 for savings : 2

  Please enter your account no. : 15100
Enter amount to deposit in account :9000
*****************************************
  Account holder name : Pankaj
  Account Type : saving
  Account number : 15100
  Total Balance : 11000
*****************************************


  1.Deposit money in account.
  2.Display the balance.
  3.Compute and deposit Interest in Saving account.
  4.Withdraw money from account.
  5.Display detail of account holder.
  6.Exit.

    Enter your choice : 4

  Enter account type 1 for current and 2 for savings : 1
  1.By Cheque.
  2.By Self.
   Enter your choice : 1

  Please enter your account no. : 13100
   Enter the amount to withdraw from acccount : 6000
  Your account has gone below minimum balance required. That is 5000.

*****************************************
  Account holder name : Amit
  Account Type : current
  Account number : 13100
  Total Balance : 3900
  Cheque Book no. From : 274001 to 274010
*****************************************


  1.Deposit money in account.
  2.Display the balance.
  3.Compute and deposit Interest in Saving account.
  4.Withdraw money from account.
  5.Display detail of account holder.
  6.Exit.

    Enter your choice : 6
PANKAJs-iMac:oops pankaj_kumar$
```

Q8.3.

Sol-

```
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <cmath>

using namespace std;

class Date{
	int dd,mm,yyyy;
public:
	void putdob(){
		cout<<"  Enter Date of Birth : "<<endl;
		cout<<"  Enter Date : ";
		cin>>dd;
		cout<<"  Enter Month : ";
		cin>>mm;
		cout<<"  Enter Year : ";
		cin>>yyyy;
	}

	void putdoj(){
		cout<<"  Enter Date of Joining : "<<endl;
		cout<<"  Enter Date : ";
		cin>>dd;
		cout<<"  Enter Month : ";
		cin>>mm;
		cout<<"  Enter Year : ";
		cin>>yyyy;
	}

	void display_dob(){
		cout<<"  Date of Birth : "<<dd<<"/"<<mm<<"/"<<yyyy<<endl;
	}

	void display_doj(){
		cout<<"  Date of Joining : "<<dd<<"/"<<mm<<"/"<<yyyy<<endl;
	}
};

class staff{
	char name[30];
	int id;
	Date dob;
	Date doj;
public:
	void putdetail(){
		cout<<"  Enter Name : ";
		cin>>name;
		cout<<"  Enter Id : ";
		cin>>id;
		dob.putdob();
		doj.putdoj();
	}

	void display(){
		cout<<"  Name : "<<name<<endl;
		cout<<"  ID : "<<id<<endl;
		dob.display_dob();
		doj.display_doj();
	}

};

class teacher:public staff{
	char subject[20];
public:
	void putdetail_t(){
		putdetail();
```

```
                cout<<"  Enter name of subject : ";
                cin>>subject;
        }

        void display_t(){
                cout<<"*****************************************"<<endl;
                cout<<"  Detail of Teacher : "<<endl;
                display();
                cout<<"  Subject : "<<subject<<endl;
                cout<<"*****************************************"<<endl<<endl;
        }
};

class officer:public staff{
        char grade;

public:
        void putdetail_o(){
                putdetail();
                cout<<"  Enter grade of officer : ";
                cin>>grade;
        }

        void display_o(){
                cout<<"*****************************************"<<endl;
                cout<<"  Detail of Officer : "<<endl;
                display();
                cout<<"  Grade : "<<grade<<endl;
                cout<<"*****************************************"<<endl<<endl;
        }
};

class typist:public staff{
        int speed;
public:
        void putdetail_typist(){
                putdetail();
                cout<<"  Enter typing speed : ";
                cin>>speed;
        }

        void display_typist(){
                display();
                cout<<"  Speed of typing : "<<speed<<endl;
        }
};

class regular:public typist{

public:
        void putdetail_typ_r(){
                putdetail_typist();
        }

        void display_typ_r(){
                cout<<"*****************************************"<<endl;
                cout<<"  Detail of regular typist : "<<endl;
                display_typist();
                cout<<"*****************************************"<<endl<<endl;
        }
};

class casual:public typist{
        float daily_wages;
public:
        void putdetail_typ_c(){
                putdetail_typist();
                cout<<"  Enter daily daily_wages : ";
                cin>>daily_wages;
        }

        void display_typ_c(){
                cout<<"*****************************************"<<endl;
                cout<<"  Detail of regular casual : "<<endl;
```

```
                display_typist();
                cout<<"  Daily Wages is : "<<daily_wages<<endl;
                cout<<"****************************************"<<endl<<endl;
        }
};

int main(){
        teacher t;
        officer o;
        regular r_t;
        casual c_t;
        cout<<"****************************************"<<endl;
        cout<<endl<<"  Enter the detail of teacher : "<<endl;
        t.putdetail_t();
        cout<<"****************************************"<<endl;
        cout<<endl<<"  Enter the detail of officer : "<<endl;
        o.putdetail_o();
        cout<<"****************************************"<<endl;
        cout<<endl<<"  Enter the detail of regular typist : "<<endl;
        r_t.putdetail_typ_r();
        cout<<"****************************************"<<endl;
        cout<<endl<<"  Enter the detail of casual typist : "<<endl;
        c_t.putdetail_typ_c();
        t.display_t();
        o.display_o();
        r_t.display_typ_r();
        c_t.display_typ_c();
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
****************************************

  Enter the detail of teacher :
  Enter Name : Pankaj
  Enter Id : 1
  Enter Date of Birth :
  Enter Date : 25011994
  Enter Month : ^C
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
****************************************

  Enter the detail of teacher :
  Enter Name : Pankaj
  Enter Id : 1
  Enter Date of Birth :
  Enter Date : 25
  Enter Month : 01
  Enter Year : 1994
  Enter Date of Joining :
  Enter Date : 12
  Enter Month : 06
  Enter Year : 2016
  Enter name of subject : Math
****************************************

  Enter the detail of officer :
  Enter Name : Amit
  Enter Id : 2
  Enter Date of Birth :
  Enter Date : 26
  Enter Month : 07
  Enter Year : 1993
  Enter Date of Joining :
  Enter Date : 23
  Enter Month : 10
  Enter Year : 2016
  Enter grade of officer : A
****************************************

  Enter the detail of regular typist :
  Enter Name : Gagan
```

```
  Enter Id : 3
  Enter Date of Birth :
  Enter Date : 20
  Enter Month : 09
  Enter Year : 1995
  Enter Date of Joining :
  Enter Date : 3
  Enter Month : 8
  Enter Year : 2015
  Enter typing speed : 45
****************************************

  Enter the detail of casual typist :
  Enter Name : Ankit
  Enter Id : 4
  Enter Date of Birth :
  Enter Date : 16
  Enter Month : 11
  Enter Year : 1991
  Enter Date of Joining :
  Enter Date : 18
  Enter Month : 03
  Enter Year : 2014
  Enter typing speed : 35
  Enter daily daily_wages : 550
****************************************
  Detail of Teacher :
  Name : Pankaj
  ID : 1
  Date of Birth : 25/1/1994
  Date of Joining : 12/6/2016
  Subject : Math
****************************************

****************************************
  Detail of Officer :
  Name : Amit
  ID : 2
  Date of Birth : 26/7/1993
  Date of Joining : 23/10/2016
  Grade : A
****************************************

****************************************
  Detail of regular typist :
  Name : Gagan
  ID : 3
  Date of Birth : 20/9/1995
  Date of Joining : 3/8/2015
  Speed of typing : 45
****************************************

****************************************
  Detail of regular casual :
  Name : Ankit
  ID : 4
  Date of Birth : 16/11/1991
  Date of Joining : 18/3/2014
  Speed of typing : 35
  Daily Wages is : 550
****************************************

PANKAJs-iMac:oops pankaj_kumar$
```

Q8.5.

Sol-

```
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <cmath>
```

```
using namespace std;

class person{
protected:
	char * name;
	int id;
public:
	void putdetail_person(const char *n, int i){
		int len = strlen(n);
		name = new char[len+1];
		strcpy(name,n);
		id = i;
	}

	void update_name(const char * n){
		int len = strlen(n);
		name = new char[len+1];
		strcpy(name,n);
	}

	void update_id(int n_id){
		id = n_id;
	}
};

class account:virtual protected person{
protected:
	int pay;
public:
	void putdetail_acc(const char * n, int i, int p){
		putdetail_person(n,i);
		pay = p;
	}

	void update_pay(int n_pay){
		pay = n_pay;
	}
};

class admin:virtual protected person{
protected:
	int experience;
public:
	void putdetail_admin(const char * n, int i, int e){
		putdetail_person(n,i);
		experience = e;
	}

	void update_name(const char * n){
		person::update_name(n);
	}

	void update_id(int id){
		person::update_id(id);
	}

	void update_exp(int n_exp){
		experience = n_exp;
	}
};

class master:protected account, protected admin{
public:
	void putdetail_m(const char * n, int i, int p, int e){
		putdetail_acc(n,i,p);
		putdetail_admin(n,i,e);
		putdetail_person(n,i);
	}

	int check_id(){
		return id;
	}
```

```
        void update_info(){
                int ch,o_id,n_id,n_pay,o_pay,n_exp,o_exp;
                char new_name[30];
                char old_name[30];
                cout<<endl<<"  Which attribute do you want to update : ";

        cout<<endl<<"  1.Name."<<endl<<"  2.ID."<<endl<<"  3.Pay."<<endl<<"  4.Experience
."<<endl;
                cout<<endl<<"  Enter your choice : ";
                cin>>ch;
                switch(ch){
                        case 1 :
                        cout<<"  Enter old name : ";
                        cin>>old_name;
                        cout<<"  Enter new name : ";
                        cin>>new_name;
                        admin::update_name(new_name);
                        break;

                        case 2 :
                        cout<<"  Enter old ID : ";
                        cin>>o_id;
                        cout<<"  Enter new ID : ";
                        cin>>n_id;
                        admin::update_id(n_id);
                        break;

                        case 3 :
                        cout<<"  Enter old Pay : ";
                        cin>>o_pay;
                        cout<<"  Enter new Pay : ";
                        cin>>n_pay;
                        account::update_pay(n_pay);
                        break;

                        case 4 :
                        cout<<"  Enter old Experience : ";
                        cin>>o_exp;
                        cout<<"  Enter new Experience : ";
                        cin>>n_exp;
                        admin::update_exp(n_exp);
                        break;
                }
        }

        void display(){
                cout<<endl<<"  Name : "<<name<<endl;
                cout<<"  ID : "<<id<<endl;
                cout<<"  Pay : "<<pay<<endl;
                cout<<"  Experience : "<<experience<<endl;
        }
};
int main(){
        int ch,i,j,m,id,pay,exp;
        char name[30];
        cout<<"  How many master do you want to create : ";
        cin>>m;
        master M[m];
        for(i=0; i<m; i++){
                cout<<endl<<"  Enter the detail of Master["<<i+1<<"] : "<<endl;
                cout<<endl<<"  Enter name of master : ";
                cin>>name;
                cout<<"  Enter ID : ";
                cin>>id;
                cout<<"  Enter pay : ";
                cin>>pay;
                cout<<"  Enter experience : ";
                cin>>exp;
                M[i].putdetail_m(name,id,pay,exp);

        }
        do{
                cout<<endl<<endl<<"  1.Update the infomation of master."<<endl;
                cout<<"  2.Display the infomation of master."<<endl;
```

```
                cout<<"  3.Exit."<<endl;
                cout<<endl<<"  Enter you choice : ";
                cin>>ch;
                switch(ch){
                        case 1 :
                        cout<<"  Enter Id of master : ";
                        cin>>id;
                        for(i=0; i<m; i++){
                                if(M[i].check_id() == id){
                                        break;
                                }
                        }
                        M[i].update_info();
                        break;

                        case 2 :
                        cout<<"  Enter Id of master : ";
                        cin>>id;
                        for(j=0; j<m; j++){
                                if(M[j].check_id() == id){
                                        break;
                                }
                        }
                        M[j].display();
                        break;

                        case 3 :
                        exit(0);
                        break;

                        default :
                        cout<<"  Please enter correct choice!!!!!..."<<endl;
                }
        }while(true);
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  How many master do you want to create : 2

  Enter the detail of Master[1] :

  Enter name of master : Pankaj
  Enter ID : 1
  Enter pay : 4
  Enter experience : 40000

  Enter the detail of Master[2] :

  Enter name of master : Amit
  Enter ID : 2
  Enter pay : 5
  Enter experience : 55000


  1.Update the infomation of master.
  2.Display the infomation of master.
  3.Exit.

  Enter you choice : 2
  Enter Id of master : 2

  Name : Amit
  ID : 2
  Pay : 5
  Experience : 55000


  1.Update the infomation of master.
  2.Display the infomation of master.
  3.Exit.
```

```
  Enter you choice : 1
  Enter Id of master : 1

  Which attribute do you want to update :
  1.Name.
  2.ID.
  3.Pay.
  4.Experience.

  Enter your choice : 4
  Enter old Experience : 40000
  Enter new Experience : 45000


  1.Update the infomation of master.
  2.Display the infomation of master.
  3.Exit.

  Enter you choice : 2
  Enter Id of master : 1

  Name : Pankaj
  ID : 1
  Pay : 4
  Experience : 45000


  1.Update the infomation of master.
  2.Display the infomation of master.
  3.Exit.

  Enter you choice : 3
PANKAJs-iMac:oops pankaj_kumar$
```

Q8.6.

Sol-

```
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <cmath>

using namespace std;

class Date{
        int dd,mm,yyyy;
public:
        void putdob(){
                cout<<"  Enter Date of Birth : "<<endl;
                cout<<"  Enter Date : ";
                cin>>dd;
                cout<<"  Enter Month : ";
                cin>>mm;
                cout<<"  Enter Year : ";
                cin>>yyyy;
        }

        void putdoj(){
                cout<<"  Enter Date of Joining : "<<endl;
                cout<<"  Enter Date : ";
                cin>>dd;
                cout<<"  Enter Month : ";
                cin>>mm;
                cout<<"  Enter Year : ";
                cin>>yyyy;
        }

        void display_dob(){
                cout<<"  Date of Birth : "<<dd<<"/"<<mm<<"/"<<yyyy<<endl;
        }
```

```
	void display_doj(){
		cout<<"  Date of Joining : "<<dd<<"/"<<mm<<"/"<<yyyy<<endl;
	}
};

class staff{
	char name[30];
	int id;
	Date dob;
	Date doj;
public:
	void putdetail(){
		cout<<"  Enter Name : ";
		cin>>name;
		cout<<"  Enter Id : ";
		cin>>id;
		dob.putdob();
		doj.putdoj();
	}

	void display(){
		cout<<"  Name : "<<name<<endl;
		cout<<"  ID : "<<id<<endl;
		dob.display_dob();
		doj.display_doj();
	}

};

class teacher{
	char subject[20];
	staff t;
public:
	void putdetail_t(){
		t.putdetail();
		cout<<"  Enter name of subject : ";
		cin>>subject;
	}

	void display_t(){
		cout<<"****************************************"<<endl;
		cout<<"  Detail of Teacher "<<endl;
		t.display();
		cout<<"  Subject : "<<subject<<endl;
		cout<<"****************************************"<<endl<<endl;
	}
};

class officer{
	char grade;
	staff o;
public:
	void putdetail_o(){
		o.putdetail();
		cout<<"  Enter grade of officer : ";
		cin>>grade;
	}

	void display_o(){
		cout<<"****************************************"<<endl;
		cout<<"  Detail of Officer "<<endl;
		o.display();
		cout<<"  Grade : "<<grade<<endl;
		cout<<"****************************************"<<endl<<endl;
	}
};

class typist{
	int speed;
	staff t;
public:
	void putdetail_typist(){
		t.putdetail();
		cout<<"  Enter typing speed : ";
```

```
                cin>>speed;
        }

        void display_typist(){
                t.display();
                cout<<"  Speed of typing : "<<speed<<endl;
        }
};

class regular:public typist{

public:
        void putdetail_typ_r(){
                putdetail_typist();
        }

        void display_typ_r(){
                cout<<"****************************************"<<endl;
                cout<<"  Detail of regular typist "<<endl;
                display_typist();
                cout<<"****************************************"<<endl<<endl;
        }
};

class casual:public typist{
        float daily_wages;
public:
        void putdetail_typ_c(){
                putdetail_typist();
                cout<<"  Enter daily daily_wages : ";
                cin>>daily_wages;
        }

        void display_typ_c(){
                cout<<"****************************************"<<endl;
                cout<<"  Detail of regular casual "<<endl;
                display_typist();
                cout<<"  Daily Wages is : "<<daily_wages<<endl;
                cout<<"****************************************"<<endl<<endl;
        }
};

int main(){
        teacher t;
        officer o;
        regular r_t;
        casual c_t;
        cout<<"****************************************"<<endl;
        cout<<endl<<"  Enter the detail of teacher : "<<endl;
        t.putdetail_t();
        cout<<"****************************************"<<endl;
        cout<<endl<<"  Enter the detail of officer : "<<endl;
        o.putdetail_o();
        cout<<"****************************************"<<endl;
        cout<<endl<<"  Enter the detail of regular typist : "<<endl;
        r_t.putdetail_typ_r();
        cout<<"****************************************"<<endl;
        cout<<endl<<"  Enter the detail of casual typist : "<<endl;
        c_t.putdetail_typ_c();
        t.display_t();
        o.display_o();
        r_t.display_typ_r();
        c_t.display_typ_c();
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
****************************************

  Enter the detail of teacher :
  Enter Name : Pankaj
  Enter Id : 1
  Enter Date of Birth :
  Enter Date : 25
  Enter Month : 01
  Enter Year : 1994
  Enter Date of Joining :
  Enter Date : 12
  Enter Month : 09
  Enter Year : 2016
  Enter name of subject : Math
****************************************

  Enter the detail of officer :
  Enter Name : Gagan
  Enter Id : 2
  Enter Date of Birth :
  Enter Date : 13
  Enter Month : 09
  Enter Year : 1993
  Enter Date of Joining :
  Enter Date : 18
  Enter Month : 02
  Enter Year : 2016
  Enter grade of officer : A
****************************************

  Enter the detail of regular typist :
  Enter Name : Amit
  Enter Id : 3
  Enter Date of Birth :
  Enter Date : 15
  Enter Month : 10
  Enter Year : 1996
  Enter Date of Joining :
  Enter Date : 17
  Enter Month : 02
  Enter Year : 2015
  Enter typing speed : 45
****************************************

  Enter the detail of casual typist :
  Enter Name : Deepak
  Enter Id : 4
  Enter Date of Birth :
  Enter Date : 26
  Enter Month : 09
  Enter Year : 1995
  Enter Date of Joining :
  Enter Date : 15
  Enter Month : 08
  Enter Year : 2015
  Enter typing speed : 40
  Enter daily daily_wages : 650
****************************************
  Detail of Teacher
  Name : Pankaj
  ID : 1
  Date of Birth : 25/1/1994
  Date of Joining : 12/9/2016
  Subject : Math
****************************************

****************************************
  Detail of Officer
  Name : Gagan
  ID : 2
```

```
  Date of Birth : 13/9/1993
  Date of Joining : 18/2/2016
  Grade : A
****************************************

****************************************
  Detail of regular typist
  Name : Amit
  ID : 3
  Date of Birth : 15/10/1996
  Date of Joining : 17/2/2015
  Speed of typing : 45
****************************************

****************************************
  Detail of regular casual
  Name : Deepak
  ID : 4
  Date of Birth : 26/9/1995
  Date of Joining : 15/8/2015
  Speed of typing : 40
  Daily Wages is : 650
****************************************

PANKAJs-iMac:oops pankaj_kumar$
```

QTest_q2.

Sol-

```
#include <iostream>
#include <cstdlib>

using namespace std;

class complex{
        float real;
        float imag;
public:
        complex(){
                real = 0.0;
                imag = 0.0;
        }
        complex(complex &c){
                real = c.real;
                imag = c.imag;
        }
        friend ostream & operator << (ostream &out, const complex &c);
        friend istream & operator >> (istream &in, complex &c);
        complex operator + ( complex &c );
        complex operator * ( complex &m );
        complex operator - ( complex &s );
        complex operator / ( complex &d );
};

ostream & operator << (ostream &out, const complex &c){
        out<<c.real<<" + "<<c.imag<<"i"<<endl;
        out<<"****************************************"<<endl<<endl;
        return out;
}

istream & operator >> (istream &in, complex &c){
        cout<<"  Enter real part: ";
        in>>c.real;
        cout<<"  Enter imaginary part: ";
        in>>c.imag;
        return in;
}

complex complex::operator + ( complex &c ){
        complex add;
        add.real = real + c.real;
```

```
        add.imag = imag + c.imag;
        return add;
}

complex complex::operator * ( complex &m ){
        complex mul;
        mul.real = (real*m.real) - (imag*m.imag);
        mul.imag = (real*m.imag) + (imag*m.real);
        return mul;
}

complex complex::operator - ( complex &s ){
        complex sub;
        sub.real = real - s.real;
        sub.imag = imag - s.imag;
        return sub;
}

complex complex::operator / ( complex &d ){
        complex div;
        div.real = ((real*d.real) + (imag*d.imag)) / ((d.real*d.real)+(d.imag*d.imag));
        div.imag = ((d.real*imag) - (real*d.imag)) / ((d.real*d.real)+(d.imag*d.imag));
        return div;
}

int main(){
        int c;
        complex c1,c2,c3;
        cin>>c1;
        cout<<endl<<"*****************************************"<<endl;
        cout<<"  First complex number is (c1): ";
        cout<<c1;
        cin>>c2;
        cout<<endl<<"*****************************************"<<endl;
        cout<<"  Second complex number is (c2): ";
        cout<<c2;
        do{
                cout<<"  1.Addition of two complex numbers."<<endl;
                cout<<"  2.Multiply of two complex numbers."<<endl;
                cout<<"  3.Subtrect one complex number form other complex number."<<endl;
                cout<<"  4.Divide one complex number by other complex number."<<endl;
                cout<<"  5.Use the copy constructor."<<endl;
                cout<<"  6.Exit."<<endl;
                cout<<endl<<"  Enter your choice: ";
                cin>>c;
                switch(c){

                        case 1:
                        c3 = c1 + c2;
                        cout<<endl<<"*****************************************"<<endl;
                        cout<<"  First complex number is (c1): ";
                        cout<<c1;
                        cout<<"  Second complex number is (c2): ";
                        cout<<c2;
                        cout<<"  Addition of c1 and c2 is: ";
                        cout<<c3;
                        break;

                        case 2:
                        c3 = c1 * c2;
                        cout<<endl<<"*****************************************"<<endl;
                        cout<<"  First complex number is (c1): ";
                        cout<<c1;
                        cout<<"  Second complex number is (c2): ";
                        cout<<c2;
                        cout<<"  Multiplication of c1 and c2 is: ";
                        cout<<c3;
                        break;

                        case 3:
                        c3 = c1 - c2;
                        cout<<endl<<"*****************************************"<<endl;
                        cout<<"  First complex number is (c1): ";
                        cout<<c1;
```

```
                        cout<<"  Second complex number is (c2): ";
                        cout<<c2;
                        cout<<"  Subtraction of c1 - c2 : ";
                        cout<<c3;
                        break;

                        case 4:
                        c3 = c1 / c2;
                        cout<<endl<<"*****************************************"<<endl;
                        cout<<"  First complex number is (c1): ";
                        cout<<c1;
                        cout<<"  Second complex number is (c2): ";
                        cout<<c2;
                        cout<<"  c1/c2 : ";
                        cout<<c3;
                        break;

                        case 5:
                        c3 = c1;
                        cout<<endl<<"*****************************************"<<endl;
                        cout<<"  First complex number is (c1): ";
                        cout<<c1;
                        cout<<"  Second complex number is (c2): ";
                        cout<<c2;
                        cout<<"  Copy constructor of (c1)  c3 : ";
                        cout<<c3;
                        break;

                        case 6:
                        cout<<"  Thank you Have a nice day......"<<endl<<endl;
                        exit(0);
                        break;
                        default:
                        cout<<"  You entered a wrong choice. Please try again leater.";

        cout<<endl<<"*********************************************************"<<endl<<end
l;
                        exit(0);
                        break;
                }
        }while(true);
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ g++ test_q2.cpp
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Enter real part: 3
  Enter imaginary part: 4

*****************************************
  First complex number is (c1): 3 + 4i
*****************************************

  Enter real part: 4
  Enter imaginary part: 5

*****************************************
  Second complex number is (c2): 4 + 5i
*****************************************

  1.Addition of two complex numbers.
  2.Multiply of two complex numbers.
  3.Subtrect one complex number form other complex number.
  4.Divide one complex number by other complex number.
  5.Use the copy constructor.
  6.Exit.

  Enter your choice: 1

*****************************************
  First complex number is (c1): 3 + 4i
*****************************************
```

```
  Second complex number is (c2): 4 + 5i
****************************************

  Addition of c1 and c2 is: 7 + 9i
****************************************

  1.Addition of two complex numbers.
  2.Multiply of two complex numbers.
  3.Subtrect one complex number form other complex number.
  4.Divide one complex number by other complex number.
  5.Use the copy constructor.
  6.Exit.

  Enter your choice: 2

****************************************
  First complex number is (c1): 3 + 4i
****************************************

  Second complex number is (c2): 4 + 5i
****************************************

  Multiplication of c1 and c2 is: -8 + 31i
****************************************

  1.Addition of two complex numbers.
  2.Multiply of two complex numbers.
  3.Subtrect one complex number form other complex number.
  4.Divide one complex number by other complex number.
  5.Use the copy constructor.
  6.Exit.

  Enter your choice: 3

****************************************
  First complex number is (c1): 3 + 4i
****************************************

  Second complex number is (c2): 4 + 5i
****************************************

  Subtraction of c1 - c2 : -1 + -1i
****************************************

  1.Addition of two complex numbers.
  2.Multiply of two complex numbers.
  3.Subtrect one complex number form other complex number.
  4.Divide one complex number by other complex number.
  5.Use the copy constructor.
  6.Exit.

  Enter your choice: 4

****************************************
  First complex number is (c1): 3 + 4i
****************************************

  Second complex number is (c2): 4 + 5i
****************************************

  c1/c2 : 0.780488 + 0.0243902i
****************************************

  1.Addition of two complex numbers.
  2.Multiply of two complex numbers.
  3.Subtrect one complex number form other complex number.
  4.Divide one complex number by other complex number.
  5.Use the copy constructor.
  6.Exit.

  Enter your choice: 5

****************************************
```

```
  First complex number is (c1): 3 + 4i
****************************************

  Second complex number is (c2): 4 + 5i
****************************************

  Copy constructor of (c1)  c3 : 3 + 4i
****************************************

  1.Addition of two complex numbers.
  2.Multiply of two complex numbers.
  3.Subtrect one complex number form other complex number.
  4.Divide one complex number by other complex number.
  5.Use the copy constructor.
  6.Exit.

  Enter your choice: 6
  Thank you Have a nice day......

PANKAJs-iMac:oops pankaj_kumar$
```

QTest_q3.

Sol-

```
#include <iostream>

using namespace std;

class result{
	char name[30];
	int roll_no;
	float marks[5];
	float per;
	char grade;
public:
	void putdata();
	void getdata();
};

void result::putdata(){
	float sum = 0.0;
	cout<<"  Enter name : ";
	cin>>name;
	cout<<"  Enter roll no : ";
	cin>>roll_no;
	cout<<"  Enter marks of five subjects : "<<endl;
	for(int i = 0; i < 5; i++){
		cout<<"  Subject "<<i+1<<" = ";
		cin>>marks[i];
		sum = sum + marks[i];
	}
	per = sum/5;
	if(per<100 && per>=90){
		grade = 'A';
	}
	else
		if(per<90 && per>=85){
			grade = 'B';
		}
		else
			if(per<85 && per>=80){
				grade = 'C';
			}
			else
				if(per<80 && per>=75){
					grade = 'D';
				}
				else
					if(per<70 && per>=70){
						grade = 'E';
					}
```

```
                                        else
                                                if(per<70){
                                                        grade = 'F';
                                                }
}

void result::getdata(){
        cout<<"  Name of the student : "<<name<<endl;
        cout<<"  Roll No. of student : "<<roll_no<<endl;
        cout<<"  Marks : "<<endl;
        for(int i = 0; i < 5; i++ ){
                cout<<"     Subject "<<i+1<<" = "<<marks[i]<<endl;
        }
        cout<<"  Percentage : "<<per<<endl<<endl;
        cout<<"  Grade : "<<grade<<endl<<endl;
}

int main(){
        int n;
        cout<<"  Enter the number of student : ";
        cin>>n;
        result r[n];
        for( int i = 0; i < n; i++ ){
        cout<<"  Student : "<<i+1<<endl;
        r[i].putdata();
        cout<<"****************************************"<<endl;
        }
        for( int i = 0; i < n; i++ ){
                cout<<endl<<"****************************************"<<endl;
                cout<<"  Detail of Student : "<<i+1<<endl<<endl;
                r[i].getdata();
                cout<<"****************************************"<<endl;
        }
        return 0;
}
```

OUTPUT

```
PANKAJs-iMac:oops pankaj_kumar$ ./a.out
  Enter the number of student : 2
  Student : 1
  Enter name : Amit
  Enter roll no : 1943
  Enter marks of five subjects :
  Subject 1 = 89
  Subject 2 = 98
  Subject 3 = 86
  Subject 4 = 87
  Subject 5 = 88
****************************************
  Student : 2
  Enter name : Gagan
  Enter roll no : 1944
  Enter marks of five subjects :
  Subject 1 = 45
  Subject 2 = 43
  Subject 3 = 42
  Subject 4 = 41
  Subject 5 = 40
****************************************

****************************************
  Detail of Student : 1

  Name of the student : Amit
  Roll No. of student : 1943
  Marks :
     Subject 1 = 89
     Subject 2 = 98
     Subject 3 = 86
     Subject 4 = 87
     Subject 5 = 88
```

```
  Percentage : 89.6

  Grade : B

****************************************

****************************************
  Detail of Student : 2

  Name of the student : Gagan
  Roll No. of student : 1944
  Marks :
     Subject 1 = 45
     Subject 2 = 43
     Subject 3 = 42
     Subject 4 = 41
     Subject 5 = 40
  Percentage : 42.2

  Grade : F

****************************************
PANKAJs-iMac:oops pankaj_kumar$
```