Q1. Define a class Person with instance members name and age.
Also define member functions setName(), setAge(), getName(), getAge().
Now define class Employee by inheriting Person class.
In the Employee class define empid and salary as instance members.
Also define setEmpid, setSalary, getEmpid, getSalary.

```
#include <iostream>
using namespace std;

class Person {

	private:

		string name;
		int age;

	public:

		void setName(void);
		void setAge(void);
		void getName(void);
		void getAge(void);
};

void Person::setName(void) {

	cout << "Enter Name : ";
	getline(cin, name);
}

void Person::setAge(void) {

	cout << "Enter Age : ";
	cin >> age;
}

void Person::getName(void) {

	cout << "Your Name : ";
	cout << name;
}

void Person::getAge(void) {

	cout << "Your Age : ";
	cout << age;
}
```

```
class Employee : public Person {

	private:

		int empid;
		float salary;

	public:

		void setEmpid(void);
		void setSalary(void);
		void getEmpid(void);
		void getSalary(void);

};

void Employee::setEmpid(void) {

	cout << "Enter Empid : ";
	cin >> empid;
}

void Employee::setSalary(void) {

	cout << "Enter Salary : ";
	cin >> salary;
}

void Employee::getEmpid(void) {

	cout << "Your Empid : ";
	cout << empid;
}

void Employee::getSalary(void) {

	cout << "Your Salary : ";
	cout << salary;
}

int main() {

	Employee e1;

	e1.setName();
	e1.setAge();
	e1.setEmpid();
	e1.setSalary();
```

```
        cout << endl;

        e1.getName();
        cout << endl;

        e1.getAge();
        cout << endl;

        e1.getEmpid();
        cout << endl;

        e1.getSalary();
}
```

Q2. Write a C++ program to add two numbers using single inheritance. Accept these two numbers from the user in base class and display the sum of these two numbers in derived class.

```
#include <iostream>
using namespace std;

class Base {

	private:

		int a, b;

	public:

		void setNum(int x, int y) {

			a = x;
			b = y;
		}

		int add(void) {

			return a + b;
		}
};

class Derive: public Base {

	public:

		void getNum(void) {

			cout << "Addition is : " << add();
		}
};

int main() {

	Derive d;

	d.setNum(5, 6);
	d.getNum();

}
```

Q3. Write a C++ program to calculate the percentage of a student using multi-level inheritance. Accept the marks of three subjects in base class. A class will be derived from the above mentioned class which includes a function to find the total marks obtained and another class derived from this class which calculates and displays the percentage of students.

```
#include <iostream>
using namespace std;

class Marks {

private:
   float math, english, science;

public:
   void Input_Marks(void);

   float s_math(void) {
      return math;
   }
   float s_english(void) {
      return english;
   }
   float s_science(void) {
      return science;
   }
};

void Marks::Input_Marks(void) {

   cout<<"Enter Marks of Following Subjects : "<<endl;

   cout<<"\tMath : ";
   cin>>math;

   cout<<"\tEnglish : ";
   cin>>english;

   cout<<"\tScience : ";
   cin>>science;
}

class Total:public Marks {

private:
   float m, e, s, total;

public:
   void Marks_Obtain(void);
   float s_total(void) {
      return total;
   }
};

void Total::Marks_Obtain(void) {
```

```
    m = s_math();
    e = s_english();
    s = s_science();

    cout<<endl;

    cout<<"Maths = "<<m<<" Out of 100"<<endl;
    cout<<"English = "<<e<<" Out of 100"<<endl;
    cout<<"Science = "<<s<<" Out of 100"<<endl;

    cout<<endl;
    total = m+e+s;
    cout<<"Total Marks : "<<total<<" Out of 300"<<endl;
}
class Calculate:public Total {
private:
    float percentage;
public:
    void disp(void) {
        cout<<endl;
        percentage = (s_total() / 300.0) * 100.0;
        cout<<"Percentage : "<<percentage;
    }
};

int main() {
    Calculate c;
    c.Input_Marks();
    c.Marks_Obtain();
    c.disp();

}
```

Q4. Write a C++ program to design a base class Person (name, address, phone_no). Derive a class Employee (eno, ename) from Person. Derive a class Manager (designation, department name, basic-salary) from Employee. Write a menu driven program to:

a. Accept all details of 'n' managers.
b. Display manager having highest salary

```
#include <iostream>
#include <conio.h>
using namespace std;

class Person {

private:
   string name, address, phone_no;

public:
   void p_name(void) {
      getline(cin, name);
   }
   void p_address(void) {
      getline(cin, address);
   }
   void p_phone_no(void) {
      getline(cin, phone_no);
   }
   string ret_name(void) {
      return name;
   }
};

class Employee:public Person {

private:
   string eno;
   string ename;

public:
   void e_eno(void) {
      getline(cin, eno);
   }
   void e_ename(void) {
      getline(cin, ename);
   }
};

class Manager:public Employee {

private:
```

```
    string designation, department_name;
    float basic_salary;

public:
    void input(void);
    float ret_sal(void) {
        return basic_salary;
    };
};

void Manager::input(void) {

fflush(stdin);
    cout<<"Enter Manager Name : ";
    p_name();

    cout<<"Enter Manager Address : ";
    p_address();

    cout<<"Enter Manager Phone No. : ";
    p_phone_no();

    cout<<"Enter Manager Employee No. :";
    e_eno();

    cout<<"Enter Manager Employee Name :";
    e_ename();

    cout<<"Enter Designation : ";
    getline(cin, designation);

    cout<<"Enter Department Name : ";
    getline(cin, department_name);

    cout<<"Enter Basic Salary : ";
    cin>>basic_salary;
}

int main() {

    int n, high = 0;

    cout<<"How many managers you want to enter : ";
    cin>>n;

    Manager m[n];

    for(int i=0; i<n; i++) {
```

```
        m[i].input();
        cout<<"---------------------------------"<<endl<<endl;
    }

    for(int i=1; i<n; i++) {

        if(m[high].ret_sal() > m[i].ret_sal())
            high = high;
        else
            high = i;
    }

    cout<<"Highest Salary : "<<m[high].ret_sal()<<endl;
    cout<<"And, Manager Name is : "<<m[high].ret_name();

}
```

Q5. Write a C++ program to define a base class Item (item-no, name, price).
Derive a class Discounted-Item (discount-percent). A customer purchases 'n' items.
Display the item-wise bill and total amount using appropriate format.

```
#include <iostream>
using namespace std;

class Item {

	private:

		int item_no;
		string name;
		float price;

	public:

		void set_item_no(void) {
			cin >> item_no;
		}

		void set_name(void) {
			getline(cin, name);
		}

		void set_price(void) {
			cin >> price;
		}

		int ret_item_no(void) {
			return item_no;
		}

		string ret_name(void) {
			return name;
		}

		float ret_price(void) {
			return price;
		}

};

class Discounted_Item: public Item {

	private:

		float discount_percent;
```

```
	public:

		void input(void);
		void disp(void);

		float ret_discount_percent(void) {
			return discount_percent;
		}
};

void Discounted_Item::input(void) {

	cout << "Enter Item name : ";
	set_name();

	cout << "Enter Item no : ";
	set_item_no();
	fflush(stdin);

	cout << "Enter Item price : ";
	set_price();
	fflush(stdin);

	cout << "Enter Discount Percent : ";
	cin >> discount_percent;
	fflush(stdin);
	cout << "---------------------------------" << endl << endl;
}

void Discounted_Item::disp(void) {

	cout << "Item Name : ";
	cout << ret_name() << endl;

	cout << "Item No. : ";
	cout << ret_item_no() << endl;

	cout << "Item Price : ";
	cout <<		ret_price() << endl;

	cout << "Discount Percent : ";
	cout << discount_percent << endl;

	cout << "Discounted Price : ";
	cout << ret_price() - ((discount_percent / 100)*ret_price()) << endl;

	cout << "---------------------------------" << endl;
}
```

```
int main() {

	int n;
	float total_price = 0.0, total_discount = 0.0;

	cout << "How many items you want to enter? : ";
	cin >> n;

	fflush(stdin);

	Discounted_Item d[n];

	for (int i = 0; i < n; i++) {

		d[i].input();
	}

	for (int i = 0; i < n; i++) {

		d[i].disp();

		total_price = d[i].ret_price() + total_price;
		total_discount = ( d[i].ret_price() - ((d[i].ret_discount_percent() / 100) * d[i].ret_price())) +
total_discount;
	}

	cout << "---------------------------------" << endl;

	cout << "Total price : " << total_price << endl;
	cout << "Total discount : " << total_discount;
}
```

Q6. Write a C++ program to demonstrate how a common friend function can be used to exchange the private values of two classes. (Use call by reference method).

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

using namespace std;

class Fun2;

class Fun1 {

	private:
		int a;

	public:
		void get(void) {
			cout << "Enter a : ";
			cin >> a;
		}

		friend void exchange(Fun1 &f1, Fun2 &f2);
		void disp(void) {
			cout << "a : " << a;
		}
};

class Fun2 {

	private:
		int b;

	public:
		void get(void) {
			cout << "Enter b : ";
			cin >> b;
		}

		void disp(void) {
			cout << "b : " << b;
		}
		friend void exchange(Fun1 &f1, Fun2 &f2);
};

void exchange(Fun1 &f1, Fun2 &f2) {

	int tmp;

	tmp = f1.a;
```

```
        f1.a = f2.b;
        f2.b = tmp;
}

int main() {


        Fun1 x;
        Fun2 y;

        x.get();
        y.get();

        exchange(x, y);

        x.disp();
        y.disp();
}
```

Q7. Write class declarations and member function definitions for a C++ base class to represent an Employee (emp-code, name). Derive two classes as Fulltime (daily rate, number of days, salary) and Parttime (number of working hours, hourly rate, salary).
Write a menu driven program to:
1. Accept the details of 'n' employees.
2. Display the details of 'n' employees.
3. Search a given Employee by emp-code.

```
#include <iostream>
#include <conio.h>
using namespace std;

class Employee {

	private:
		int emp_code;
		string name;

	public:
		void set_employee_data(void) {
			cout << "Enter emp code : ";
			cin >> emp_code;
			cin.ignore();

			cout << "Enter name : ";
			getline(cin, name);
		}

		void emp_disp(void) {

			cout << "Emp code : " << emp_code << endl;

			cout << "Name : " << name << endl;
		}

		int code(void){
			return emp_code;
		}

};

class Fulltime: public Employee {

	private:
		float daily_rate, salary, number_of_days;

	public:

		void set_fulltime_data(void) {
```

```
                cout << "Enter daily rate : ";
                cin >> daily_rate;

                cout << "Enter number of days : ";
                cin >> number_of_days;

                salary = daily_rate * number_of_days;
        }

        void Full_disp(void) {
                cout << "Daily rate : " << daily_rate << endl;

                cout << "Salary : " << salary << endl;

                cout << "Number of days : " << number_of_days << endl;
        }
};

class Parttime: public Employee {

        private:
                float number_of_working_hours, hourly_rate, salary;

        public:

                void set_parttime_data(void) {
                        cout << "Enter number of working hours : ";
                        cin >> number_of_working_hours;

                        cout << "Enter hourly rate : ";
                        cin >>  hourly_rate;

                        salary = number_of_working_hours * hourly_rate;
                }

                void Part_disp(void) {

                        cout << "Number of working hours : " << number_of_working_hours << endl;

                        cout << "Hourly rate : " << hourly_rate << endl;

                        cout << "Salary : " << salary << endl;
                }
};

int main() {

        int choice, fn = -1, pn = -1, FN, PN;
```

```
cout<<"How many Fulltime employee you want to enter : ";
cin>>FN;

cout<<"How many Parttime employee you want to enter : ";
cin>>PN;

Fulltime f[FN];
Parttime p[PN];


do {
	system("cls");

	cout<<"1. Enter record : "<< endl;
	cout<<"2. Display record : "<<endl;
	cout<<"3. Search record : "<<endl;
	cout<<"4. Exit : "<<endl;
	cout<<"Enter choice : ";
	cin>>choice;

	switch (choice) {

		case 1: {
			system("cls");
			int tmp;

			cout << "1. Fulltime employee" << endl;
			cout << "2. Parttime employee" << endl << endl;

			cout << "Enter your choice : ";
			cin >> tmp;

			if (tmp == 1) {
				system("cls");

				if( (fn = fn + 1) < FN){
					f[fn].set_employee_data();
					f[fn].set_fulltime_data();

					cout<<"Data saved";
					getch();
				}
				else{
					cout<<"Fulltime employee memory full";
					getch();
				}

			} else if (tmp == 2) {
				if( (pn = pn + 1) < PN){
```

```
                        system("cls");

                        p[pn].set_employee_data();
                        p[pn].set_parttime_data();

                        cout<<"Data saved";
                        getch();
                }
                else{
                        cout<<"Parttime employee memory full";
                        getch();
                }
        } else {
                cout << "Wrong choice";
                getch();
        }
        break;
}
case 2: {
        system("cls");

        for(int i = 0; i < FN ; i++){
                f[i].emp_disp();
                f[i].Full_disp();

                cout<<endl<<"-------------------------------------"<<endl;
        }

        for(int i = 0; i < FN ; i++){
                p[i].emp_disp();
                p[i].Part_disp();

                cout<<endl<<"-------------------------------------"<<endl;
        }
        getch();
        break;
}
case 3:{
        system("cls");
        int code, flag=0;

        cout<<"Enter employee code : ";
        cin>>code;

        for(int i=0; i < FN ; i++){
                if(code == f[i].code()){
                        f[i].emp_disp();
                        f[i].Full_disp();
```

```
                                        cout<<endl<<"-------------------------------------"<<endl;
                                        getch();
                                        flag=1;
                                        break;
                                }
                        }
                        if(flag == 0){
                                for(int i=0; i < PN ; i++){
                                        if(code == p[i].code()){
                                                p[i].emp_disp();
                                                p[i].Part_disp();

                                                cout<<endl<<"-------------------------------------"<<endl;
                                                getch();
                                                flag=1;
                                                break;
                                        }
                                }
                        }
                        if(flag == 0){
                                cout<<"Data not found";
                                getch();
                        }
                        break;
                }
                default :{
                        if(choice == 4){
                                cout<<"Programm exited";
                                getch();
                        }
                        else{
                                cout<<"Enter wrong choice";
                                getch();
                        }
                }
        }
    } while (choice != 4);
}
```

Q8. In a bank, different customers have savings account. Some customers may have taken a loan from the bank. So bank always maintain information about bank depositors and borrowers. Design a Base class Customer (name, phone-number). Derive a class Depositor(accno, balance) from Customer. Again, derive a class Borrower (loan-no, loan-amt) from Depositor. Write necessary member functions to read and display the details of 'n' customers.

```
#include <iostream>
#include <conio.h>
using namespace std;

class Customer{
	private:
		string name, phone_no;

	public:
		void setCustdata(void){
			cout<<"Enter name : ";
			getline(cin, name);

			cout<<"Enter phone number : ";
			getline(cin, phone_no);
		}

		void dispCustdata(void){
			cout<<"Name : "<<name<<endl;
			cout<<"Phone number : "<<phone_no<<endl;
		}
};

class Depositor: public Customer{
	private:
		string accno;
		float balance;

	public:
		void setDepodata(void){
			cout<<"Enter accoutn number : ";
			getline(cin, accno);

			cout<<"Enter balance : ";
			cin>>balance;
		}

		void dispDepodata(void){
			cout<<"Account number : "<<accno<<endl;
			cout<<"Balance : "<<balance<<endl;
		}
};

class Borrower: public Depositor{
	private:
		int loan_no;
		float loan_amt;

	public:
		void setBordata(void){
```

```
                cout<<"Enter loan no : ";
                cin>>loan_no;

                cout<<"Enter loan amt : ";
                cin>>loan_amt;
        }

        void dispBordata(void){
                cout<<"Loan no : "<<loan_no<<endl;
                cout<<"Loan amt : "<<loan_amt<<endl;
        }
};

int main() {

        int n;

        cout<<"How many customer you want to enter : ";
        cin>>n;

        system("cls");

        Borrower b[n];

        for(int i = 0; i < n; i++){
                cin.ignore();

                b[i].setCustdata();
                b[i].setDepodata();
                b[i].setBordata();

                cout<<"--------------------------------------"<<endl<<endl;
        }

        for(int i = 0; i < n; i++){
                cout<<"Details of custormer"<<endl<<endl;

                b[i].dispCustdata();
                b[i].dispDepodata();
                cout<<"--------------------------------------"<<endl<<endl;

                b[i].dispBordata();
                cout<<"--------------------------------------"<<endl<<endl;
        }

        getch();
}
```

Q9.Write a C++ program to implement the following class hierarchy:
Student: id, name
StudentExam (derived from Student): Marks of 6 subjects
StudentResult (derived from StudentExam) : percentage
Define appropriate functions to accept and display details.
Create 'n' objects of the StudentResult class and display the marklist.

```
#include <iostream>
#include <conio.h>
using namespace std;


class Student{
	private:
		int id;
		string name;


	public:
		void setStudata(void){
			cout<<"Enter id : ";
			cin>>id;
			cin.ignore();


			cout<<"Enter name : ";
			getline(cin, name);
		}


		void dispStudata(void){
			cout<<"Id : "<<id<<endl;
			cout<<"Name : "<<name<<endl;
		}
};
class StudentExam: public Student{
		private:
			float mark[6];
```

```
        public:
            void setMarkdata(void){
                for(int i=0; i < 6; i++){
                    cout<<"Enter Subject "<<i+1<<" : ";
                    cin>>mark[i];
                }
            }

            void dispMarkdata(void){
                for(int i=0; i < 6; i++){
                    cout<<"Subject "<<i+1<<" : "<<mark[i]<<endl;
                }
            }

            float retMark(int i){
                return mark[i];
            }
};

class StudentResult: public StudentExam{
    private:
        float percentage, total=0;

    public:
        void dispResult(void){
            for(int i = 0; i < 6; i++){
                total = total + retMark(i);
            }
            percentage = total/6;
```

```
                cout<< percentage;
        }
};
int main() {
    int n;

    cout<<"Enter number of student you want : ";
    cin>>n;
    cout<<"-----------------------------------------------"<<endl;

    StudentResult s[n];

    for(int i=0; i < n; i++){
        s[i].setStudata();
        s[i].setMarkdata();
    }

    cout<<"-----------------------------------------------"<<endl;
    cout<<"      ********Student Marksheet********       "<<endl;
    cout<<"-----------------------------------------------"<<endl;

    for(int i=0; i < n; i++){
        s[i].dispStudata();
        cout<<endl;

        s[i].dispMarkdata();
        cout<<endl;

        cout<<"Total Percentage : ";
```

```
            s[i].dispResult();
            cout<<endl<<"-----------------------------------------"<<endl;
        }
        getch();
}
```

Q10. Consider two base classes
worker(int code, char name, float salary),
officer(float DA, HRA)
class manger(float TA(is 10% of salary), gross salary) is derived from both base classes.
Write necessary member functions.

```
#include <iostream>
#include <conio.h>
using namespace std;

class Worker{
	private:
		int code;
		char name[50];
		float salary;

	public:
		void setWorker(void){
			cout<<"Enter code : ";
			cin>>code;
			cin.ignore();

			cout<<"Enter name : ";
			gets(name);

			cout<<"Enter salary : ";
			cin>>salary;
		}

		void dispWorker(void){
			cout<<"Code : "<<code<<endl;
			cout<<"Name : "<<name<<endl;
			cout<<"Salary : "<<salary<<endl;
		}

		float retSal(void){
			return salary;
		}
};

class Officer{
	private:
		float DA, HRA;

	public:
		void setOfficer(void){
			cout<<"Enter DA : ";
			cin>>DA;
```

```
                cout<<"Enter HRA : ";
                cin>>HRA;
        }

        void dispOfficer(void){
                cout<<"DA : "<<DA<<endl;
                cout<<"HRA : "<<HRA<<endl;
        }

        float retDA_HRA(void){
                return DA+HRA;
        }
};

class Manager: public Worker, public Officer{
        private:
                float TA, Gross_sal;

        public:
                void setMangaer(void){
                        TA = (retSal() / 100) * 10;
                        Gross_sal = TA + retDA_HRA()+retSal();
                }

                void dispManager(void){
                        cout<<"TA : "<<TA<<endl;
                        cout<<"Gross salary : "<<Gross_sal<<endl;
                }
};

int main() {

        Manager m;

        cout<<"---------------------------------"<<endl;
        m.setWorker();
        m.setOfficer();
        m.setMangaer();
        cout<<"---------------------------------"<<endl;
        cout<<"       Manager Information       "<<endl;
        cout<<"---------------------------------"<<endl;
        m.dispWorker();
        m.dispOfficer();
        m.dispManager();

        getch();

}
```