Database Management System.

-supervised by:

Prof. Kaushik Majumdar,

Prof. Mihir Singh, and

Prof. Subhanjan Sarkar

Rohit Das

B. Tech(Computer Sc. and Engg)

Roll: 30000114022

Regn. No.:143000110023

6th Semester,2016

Maulana Abul Kalam Azad University of Technology,

West Bengal.

LaTeX 2017

INDEX

Sl. No.	Particulars	Date	Page No.	Signature
1.	Program to calculate average of cricket players.	28/9/16	3	
2.	Program to calculate cost for consumer.	28/9/16	4	
3.	Program to display votes received in election.	28/9/16	5	
4.	Program for operation on array elements.	19/10/16	6	
5.	Program to calculate factorial of a number.	19/10/16	7	
6.	Program to calculate sum of series1+22+32+42+....	19/10/16	7	
7.	Program to add values of Complex no. objects.	9/11/16	8	
8.	Program to convert distances to inches and add.	28/9/16	9	
9.	Program for String class with different costructors.	9/11/16	10	
10.	Program to display unique roll of Student objects.	9/11/16	10	
11.	Program to swap values of objects using 'friend' function.	9/11/16	11	
12.	Program for operation on matrices.	16/11/16	12	
13.	Program to use overloaded operators on float numbers.	16/11/16	14	
14.	Program to compare two strings by '==' operator.	16/11/16	15	
15.	Program to implement current and savings a/c.	23/11/16	16	
16.	Previous program rewritten using constructors.	23/11/16	18	
17.	Program for an educational institution.	23/11/16	20	
18.	Program to implement a network.	23/11/16	23	
19.	Program to show overriding using 'virtual'.	30/11/16	24	
20.	Program to implement Shape class using 'virtual'.	30/11/16	25	
21.	Previous program rewritten without using 'virtual'.	30/11/16	26	
22.	Program to use template for an array.	7/12/16	27	
23.	Program to implement exception handling with multiple catch.	7/12/16	28	
24.	Program to implement rethrowing an exception.	7/12/16	29	
25.	Program to implement a generic vector.	27/12/16	29	

1. Date: 20/4/2017

ID	TEAMNAME	COACH
POL	Poland	Franciszek Smuda
RUS	Russia	Dick Advocaat
CZE	Czech Republic	Michal Blek
GRE	Greece	Fernando Santos
NED	Netherlands	Bert van Marwijk
DEN	Denmark	Morten Olsen
GER	Germany	Joachim Lw
POR	Portugal	Paulo Bento
ESP	Spain	Vicente del Bosque
ITA	Italy	Cesare Prandelli
IRL	Republic of Ireland	Giovanni Trapattoni
CRO	Croatia	Slaven Bilic
UKR	Ukraine	Oleh Blokhin
SWE	Sweden	Erik Hamrn
ENG	England	Roy Hodgson
FRA	France	Laurent Blanc

ETeam (PK: ID)

ID	MDATE	STADIUM	TEAM1	TEAM2
1001	06/08/12	National Stadium (Warsaw)	POL	GRE
1002	06/08/12	Stadion Miejski (Wroclaw)	RUS	CZE
1003	06/12/12	Stadion Miejski (Wroclaw)	GRE	CZE
1004	06/12/12	National Stadium (Warsaw)	POL	RUS
1005	06/16/12	Stadion Miejski (Wroclaw)	CZE	POL
1006	06/16/12	National Stadium (Warsaw)	GRE	RUS
1007	06/09/12	Metalist Stadium	NED	DEN
1008	06/09/12	Arena Lviv	GER	POR
1009	06/13/12	Arena Lviv	DEN	POR
1010	06/13/12	Metalist Stadium	NED	GER
1011	06/17/12	Metalist Stadium	POR	NED
1012	06/17/12	Arena Lviv	DEN	GER
1013	06/10/12	PGE Arena Gdansk	ESP	ITA
1014	06/10/12	Stadion Miejski (Poznan)	IRL	CRO
1015	06/14/12	Stadion Miejski (Poznan)	ITA	CRO
1016	06/14/12	PGE Arena Gdansk	ESP	IRL
1017	06/18/12	PGE Arena Gdansk	CRO	ESP
1018	06/18/12	Stadion Miejski (Poznan)	ITA	IRL
1019	06/11/12	Donbass Arena	FRA	ENG
1020	06/11/12	Olimpiyskiy National Sports Complex	UKR	SWE
1021	06/15/12	Donbass Arena	UKR	FRA
1022	06/15/12	Olimpiyskiy National Sports Complex	SWE	ENG
1023	06/19/12	Donbass Arena	ENG	UKR
1024	06/19/12	Olimpiyskiy National Sports Complex	SWE	FRA
1025	06/21/12	National Stadium (Warsaw)	CZE	POR
1026	06/22/12	PGE Arena Gdansk	GER	GRE
1027	06/23/12	Donbass Arena	ESP	FRA
1028	06/24/12	Olimpiyskiy National Sports Complex	ENG	ITA
1029	06/27/12	Donbass Arena	POR	ESP
1030	06/28/12	National Stadium (Warsaw)	GER	ITA
1031	07/01/12	Olimpiyskiy National Sports Complex	ESP	ITA

Game(PK:ID)

MATCHID	TEAMID	PLAYER	GTIME
1001	POL	Robert Lewandowski	17
1001	GRE	Dimitris Salpingidis	51
1002	RUS	Alan Dzagoev	15
1002	RUS	Alan Dzagoev	79
1002	RUS	Roman Shirokov	24
1002	RUS	Roman Pavlyuchenko	82
1002	CZE	Vclav Pilar	52
1003	GRE	Theofanis Gekas	53
1003	CZE	Petr Jircek	3
1003	CZE	Vclav Pilar	6
1004	POL	Jakub Blaszczykowski	57
1004	RUS	Alan Dzagoev	37
1005	CZE	Petr Jircek	72
1006	GRE	Giorgos Karagounis	45
1007	DEN	Michael Krohn-Dehli	24
1008	GER	Mario Gmez	72
1009	DEN	Nicklas Bendtner	41
1009	DEN	Nicklas Bendtner	80
1009	POR	Pepe (footballer born 1983)	24
1009	POR	Hlder Postiga	36
1009	POR	Silvestre Varela	87
1010	NED	Robin van Persie	73
1010	GER	Mario Gmez	24
1010	GER	Mario Gmez	38
1011	POR	Cristiano Ronaldo	28
1011	POR	Cristiano Ronaldo	74
1011	NED	Rafael van der Vaart	11
1012	DEN	Michael Krohn-Dehli	24
1012	GER	Lukas Podolski	19
1012	GER	Lars Bender	80
1013	ESP	Cesc Fbregas	64
1013	ITA	Antonio Di Natale	61
1014	IRL	Sean St Ledger	19
1014	CRO	Mario Mandukic	3
1014	CRO	Mario Mandukic	49
1014	CRO	Nikica Jelavic	43
1015	ITA	Andrea Pirlo	39
1015	CRO	Mario Mandukic	72
1016	ESP	Fernando Torres	4
1016	ESP	Fernando Torres	70
1016	ESP	David Silva	49
1016	ESP	Cesc Fbregas	83
1017	ESP	Jess Navas	88
1018	ITA	Antonio Cassano	35
1018	ITA	Mario Balotelli	90
1019	FRA	Samir Nasri	39
1019	ENG	Joleon Lescott	30
1020	UKR	Andriy Shevchenko	55
1020	UKR	Andriy Shevchenko	62
1020	SWE	Zlatan Ibrahimovic	52

Goal (FK: MATCHID)

1..1 Show player name, team id,stadium and mdate for every german player.

Query:

```
select Goal.Player, Goal.TeamID, Game.Stadium, Game.MDate
from Goal join Game on Goal.MATCHID=Game.ID
where goal.teamid='GER'
```

Output:

PLAYER	TEAMID	STADIUM	MDATE
Mario Gmez	GER	Arena Lviv	06/09/12
Mario Gmez	GER	Metalist Stadium	06/13/12
Mario Gmez	GER	Metalist Stadium	06/13/12
Lars Bender	GER	Arena Lviv	06/17/12
Lukas Podolski	GER	Arena Lviv	06/17/12

1..2 Show team1,team2, and player for every goal scored by a player called 'Mario'.

Program:

```
select Game.team1, Game.team2, Goal.Player
from Goal join Game on Goal.MATCHID=Game.ID
where goal.matchid=game.id and goal.player like 'Mario%'
```

Output:

TEAM1	TEAM2	PLAYER
GER	POR	Mario Gmez
NED	GER	Mario Gmez
NED	GER	Mario Gmez
IRL	CRO	Mario Mandukic
IRL	CRO	Mario Mandukic
ITA	CRO	Mario Mandukic
ITA	IRL	Mario Balotelli

.

1..3 Show player, teamid, coach, gtime for all goals scored in the first 10 minutes.

Query:

```
select Goal.Player, Goal.TeamID,eteam.coach,goal.gtime
from Goal join eteam on Goal.teamID=ETeam.ID
where goal.gtime <= 10
```

Output:

PLAYER	TEAMID	COACH	GTIME
Vclav Pilar	CZE	Michal Blek	6
Petr Jircek	CZE	Michal Blek	3
Fernando Torres	ESP	Vicente del Bosque	4
Mario Mandukic	CRO	Slaven Bilic	3

.

1..4 List the dates of the matches and the name of the team in which 'Fernando Santos' was the team1 coach.

Query:

```
select Game.mdate, ETeam.teamname
from Game join eteam on Game.Team1=ETeam.ID
where coach='Fernando Santos'
```

Output:

MDATE	TEAMNAME
06/12/12	Greece
06/16/12	Greece

.

1..5 List the player for every goal scored in a game whre the stadium was 'National Stadium, Warsaw'.

Query:

```
select Goal.player
from Game join goal on Goal.matchid=game.ID
where stadium='National Stadium, Warsaw'
```

Output:

PLAYER
Robert Lewandowski
Dimitris Salpingidis
Jakub Blaszczykowski
Alan Dzagoev
Giorgos Karagounis

.

1..6 Show the name of the player who scored a goal against Germany('GER').

Query:

```
select goal.player from goal
join game on goal.matchid=game.id
where (game.team2='GER' or game.team1='GER') and goal.teamid!='GER'
```

Output:

PLAYER
Robin van Persie
Michael Krohn-Dehli

1..7 Show the stadium and the number of goals scored in each stadium. Use count().

Query:

```
select game.stadium,count(*) as goal_count from game
join goal on goal.matchid=game.id
group by game.stadium
order by count(*) asc
```

Output:

STADIUM	GOALS
Donbass Arena	2
Olimpiyskiy National Sports Complex	3
Metalist Stadium	7
PGE Arena Gdansk	7
Stadion Miejski (Poznan)	8
Stadion Miejski (Wroclaw)	9
Arena Lviv	9

.

2. Date: 27/4/2017

ID	TEAMNAME	COACH
POL	Poland	Franciszek Smuda
RUS	Russia	Dick Advocaat
CZE	Czech Republic	Michal Blek
GRE	Greece	Fernando Santos
NED	Netherlands	Bert van Marwijk
DEN	Denmark	Morten Olsen
GER	Germany	Joachim Lw
POR	Portugal	Paulo Bento
ESP	Spain	Vicente del Bosque
ITA	Italy	Cesare Prandelli
IRL	Republic of Ireland	Giovanni Trapattoni
CRO	Croatia	Slaven Bilic
UKR	Ukraine	Oleh Blokhin
SWE	Sweden	Erik Hamrn
ENG	England	Roy Hodgson
FRA	France	Laurent Blanc
ARG	Argentina	Diego Maradona

ETeam (PK: ID)

.

.

ID	MDATE	STADIUM	TEAM1	TEAM2
1001	06/08/12	National Stadium (Warsaw)	POL	GRE
1002	06/08/12	Stadion Miejski (Wroclaw)	RUS	CZE
1003	06/12/12	Stadion Miejski (Wroclaw)	GRE	CZE
1004	06/12/12	National Stadium (Warsaw)	POL	RUS
1005	06/16/12	Stadion Miejski (Wroclaw)	CZE	POL

Game(PK:ID)

ID	MDATE	STADIUM	TEAM1	TEAM2
1006	06/16/12	National Stadium (Warsaw)	GRE	RUS
1007	06/09/12	Metalist Stadium	NED	DEN
1008	06/09/12	Arena Lviv	GER	POR
1009	06/13/12	Arena Lviv	DEN	POR
1010	06/13/12	Metalist Stadium	NED	GER
1011	06/17/12	Metalist Stadium	POR	NED
1012	06/17/12	Arena Lviv	DEN	GER
1013	06/10/12	PGE Arena Gdansk	ESP	ITA
1014	06/10/12	Stadion Miejski (Poznan)	IRL	CRO
1015	06/14/12	Stadion Miejski (Poznan)	ITA	CRO
1016	06/14/12	PGE Arena Gdansk	ESP	IRL
1017	06/18/12	PGE Arena Gdansk	CRO	ESP
1018	06/18/12	Stadion Miejski (Poznan)	ITA	IRL
1019	06/11/12	Donbass Arena	FRA	ENG
1020	06/11/12	Olimpiyskiy National Sports Complex	UKR	SWE
1021	06/15/12	Donbass Arena	UKR	FRA
1022	06/15/12	Olimpiyskiy National Sports Complex	SWE	ENG
1023	06/19/12	Donbass Arena	ENG	UKR
1024	06/19/12	Olimpiyskiy National Sports Complex	SWE	FRA
1025	06/21/12	National Stadium (Warsaw)	CZE	POR
1026	06/22/12	PGE Arena Gdansk	GER	GRE
1027	06/23/12	Donbass Arena	ESP	FRA
1028	06/24/12	Olimpiyskiy National Sports Complex	ENG	ITA

.

.

MATCHID	TEAMID	PLAYER	GTIME
1001	POL	Robert Lewandowski	17
1001	GRE	Dimitris Salpingidis	51
1002	RUS	Alan Dzagoev	15
1002	RUS	Alan Dzagoev	79
1002	RUS	Roman Shirokov	24
1002	RUS	Roman Pavlyuchenko	82
1002	CZE	Vclav Pilar	52
1003	GRE	Theofanis Gekas	53
1003	CZE	Petr Jircek	3
1003	CZE	Vclav Pilar	6
1004	POL	Jakub Blaszczykowski	57
1004	RUS	Alan Dzagoev	37
1005	CZE	Petr Jircek	72
1006	GRE	Giorgos Karagounis	45
1007	DEN	Michael Krohn-Dehli	24
1008	GER	Mario Gmez	72
1009	DEN	Nicklas Bendtner	41
1009	DEN	Nicklas Bendtner	80
1009	POR	Pepe (footballer born 1983)	24
1009	POR	Hlder Postiga	36
1009	POR	Silvestre Varela	87
1010	NED	Robin van Persie	73
1010	GER	Mario Gmez	24

Goal (FK: MATCHID)

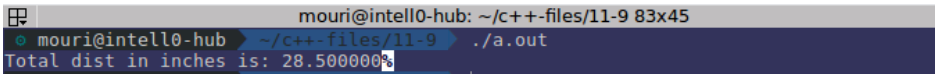
MATCHID	TEAMID	PLAYER	GTIME
1010	GER	Mario Gmez	38
1011	POR	Cristiano Ronaldo	28
1011	POR	Cristiano Ronaldo	74
1011	NED	Rafael van der Vaart	11
1012	DEN	Michael Krohn-Dehli	24
1012	GER	Lukas Podolski	19
1012	GER	Lars Bender	80
1013	ESP	Cesc Fbregas	64
1013	ITA	Antonio Di Natale	61
1014	IRL	Sean St Ledger	19
1014	CRO	Mario Mandukic	3
1014	CRO	Mario Mandukic	49
1014	CRO	Nikica Jelavic	43
1015	ITA	Andrea Pirlo	39
1015	CRO	Mario Mandukic	72
1016	ESP	Fernando Torres	4
1016	ESP	Fernando Torres	70
1016	ESP	David Silva	49
1016	ESP	Cesc Fbregas	83
1017	ESP	Jess Navas	88
1018	ITA	Antonio Cassano	35
1018	ITA	Mario Balotelli	90
1019	FRA	Samir Nasri	39
1019	ENG	Joleon Lescott	30

2..1 Show the matchdate,stadium,player name,goal time for match ids 1019 and 1021.

Query:

```
select game.mdate,game.stadium,goal.player, goal.gtime
from Game left join goal
on Goal.matchid=game.id
where game.id=1019 or game.id=1021
```

Output:



.

2..2 Show all the goal time,player, stadium where first team was Germany.

Query:

```
select goal.gtime,goal.player,game.stadium
from Game left join goal
on Goal.matchid=game.id
where game.team1='GER'
```

Output:

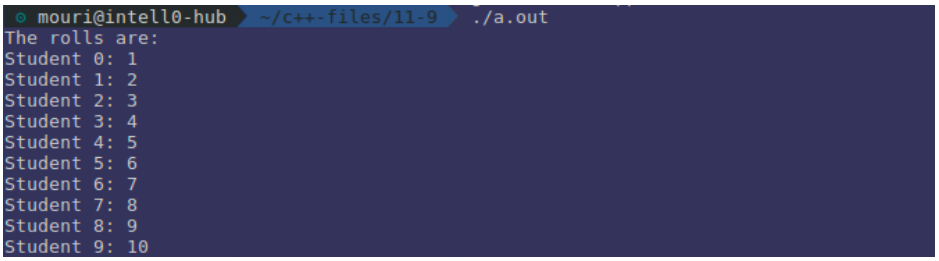
.

2..3 Show the matchdate,stadium,teamname where Diego Maradona was the coach. Use right join.

Query:

```
select game.mdate,game.stadium,eteam.teamname
from Game right join eteam
on game.team1=eteam.id or game.team2=eteam.id
where eteam.coach='Diego Maradona'
```

Output:



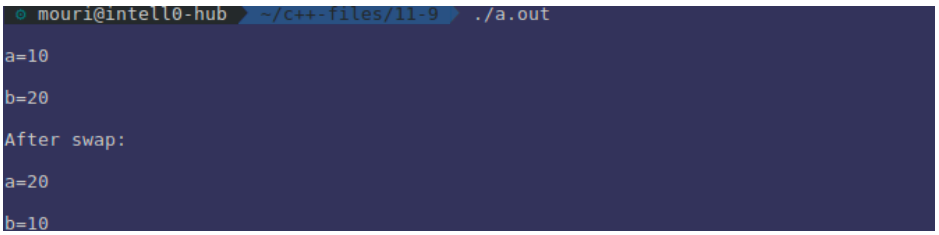
.

2..4 Show id,matchdate,stadium,coach and team name for all team2. Use full outer join.

Query:

```
select distinct game.id,game.mdate,game.stadium,eteam.coach,eteam.teamname
from Game full outer join eteam
on game.team2=eteam.id
```

Output:



.

3. Date: 27/4/2017

ORDER-ID	CUSTOMER-ID	ORDER-DATE
1	7000	4/18/2016
2	5000	4/18/2016
3	8000	4/19/2016
4	4000	4/20/2016
5	NULL	5/1/2016

Order (PK:ORDER-ID, FK:CUSTOMER-ID)

CUSTOMER-ID	LAST-NAME	FIRST-NAME	FAVORITE-WEBSITE
4000	Jackson	Joe	techonthenet.com
5000	Smith	Jane	digminecraft.com
6000	Ferguson	Samantha	bigactivities.com
7000	Reynolds	Allen	checkyourmath.com
8000	Anderson	Paige	NULL
9000	Johnson	Derek	techonthenet.com

Game(PK:CUSTOMER-ID)

EMPID	NAME	MANAGER
1	Joy Bannerjee	4
2	Rahul Das	6
3	Binoy Paul	2
4	Palash Roy	5
5	Prasanna Santra	7
6	Ananya Majumder	4
7	Albert	5

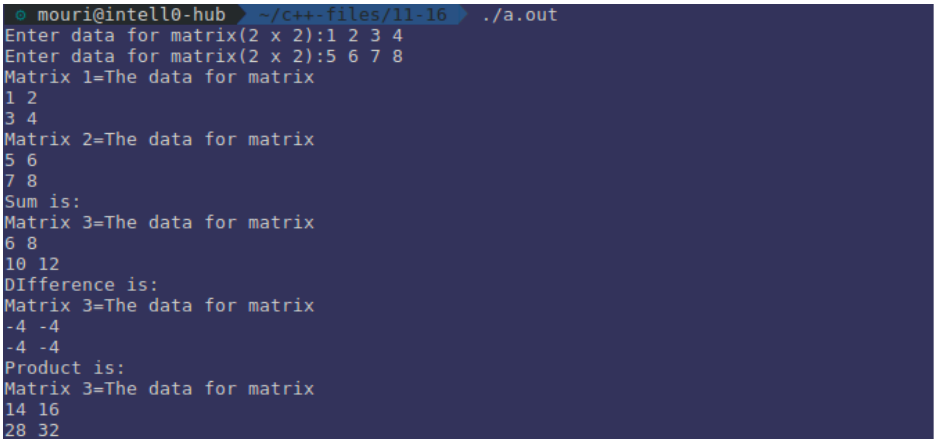
Employee (PK: EMPID)

3..1 Show employee name and Manager name for Employee ID 4 and 6.

Query:

```
select b.name as Employee, a.name as Manager
from employee a,employee b
where a.empid=b.manager and (b.empid=4 or b.empid=6)
```

Output:



.

3..2 Show the entire list of employee name and employee id under manager "Palash Roy".

Query:

```
select distinct emp.name as Employee, man.name as Manager
from employee emp,employee man
where emp.manager=man.empid and man.empid=4
```

Output:

```
mouri@intel0-hub ~/c++-files/11-16 ./a.out
f1=1.500000
f2=2.500000
Sum is:
f3=4.000000
Difference is:
f3=-1.000000
Product is:
f3=3.750000
Quotient is:
f3=0.600000
```

.

4. Write a program to compare two strings by overloading the '==' operator.

Query:

```
#include<iostream>
#include<cstdio>
#include<string>
using namespace std;
class String{
        string s;
        public:
        String(){       s="";   }
        String(string _s){      s=_s;   }
        string operator==(String);
};
string String::operator==(String str){
        if(s==str.s)return "Equals\n";
        else return "Not equals\n";
}
int main(){
        String s1("Hello"),s2("Hola"),s3("Hello");
        cout<<"Comparing Hello and Hola:"<<endl;
        cout<<(s1==s2);
        cout<<"Comparing Hello and Hello:"<<endl;
        cout<<(s1==s3);
}
```

Output:

```
mouri@intel0-hub ~/c++-files/11-16 ./a.out
Comparing Hello and Hola:
Not equals
Comparing Hello and Hello:
Equals
```

.

5. Write a program to create a class account that stores customer name, account number and type of account. Create two more classes for current a/c and savings a/c. The current a/c will have:
∗cheque facility,
∗ minimum balance and deduction for balance below that,
∗deposit and withdrawal. The saving a/c will have similar member methods, except for cheque and minimum balance, it will have an interest calculation. Do not use constructors.

Query:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
class Account{
    string customer_name,acct_type;
    ll acct;
};
class Cur_acct:public Account{
    double balance,minBal;
    public:
        void setbal(double a,double b)  {
            balance=a;
            minBal=b;
        }
        void withdrawal(int n)  {
            if(balance<minBal)balance-=(balance*0.10);
            else if(n>balance)printf("Insufficient balance.\n");
            else balance-=n;
        }
        void deposit(int n){    balance+=n;     }
        void cheque(){ cout<<"Deposited by cheque..."<<endl;    }
        void show(){
            printf("Current balance=%.2f\n",balance);
        }
};
class Sav_acct:public Account{
    double balance;
    public:
        void setbal(double a){    balance=a;    }
        void withdrawal(int n){
            if(n>balance)printf("Insufficient balance.\n");
            else balance-=n;
        }
        void deposit(int n){    balance+=n;     }
        void interest(){    balance+=(balance*0.20);    }
        void show(){
            printf("Current balance:%.2f\n",balance);
        }
};
int main(){
    cout<<"Enter type of account(s for savings, c for current):"<<endl;
    char c;    cin>>c;
    cout<<"Enter balance:"<<endl;
    int bal;    cin>>bal;
    if(c=='c'){
                cout<<"Enter minimum balance:"<<endl;
                int minbal;     cin>>minbal;
```

```
                Cur_acct c1;
                c1.setbal(bal,minbal);
                cout<<"Enter amount to deposit:"<<endl;
                int n;  cin>>n;
                cout<<"Do you want to use cheque or cash?(c for cheque)..
                "<<endl;
                char c; cin>>c;
                if(c=='c')c1.cheque();
                c1.deposit(n);  cout<<"Deposited: Rs."<<n<<"\n";
                c1.show();
                cout<<"Enter amount to withdraw:"<<endl;
                cin>>n; c1.withdrawal(n);
                cout<<"Withdrawal: Rs."<<n<<"\n";       c1.show();
                cout<<"Enter amount to withdraw:"<<endl;
                cin>>n; c1.withdrawal(n);
                cout<<"Withdrawal: Rs."<<n<<"\n";       c1.show();
        }
    else{
                Sav_acct s1;
                s1.setbal(bal);
                cout<<"Enter amount to deposit:"<<endl;
                int n;  cin>>n;
                s1.deposit(n);
                cout<<"Deposited: Rs."<<n<<"\n";        s1.show();
                cout<<"Enter amount to withdraw:"<<endl;
        cin>>n; s1.withdrawal(n);
                cout<<"Withdrawal: Rs."<<n<<"\n";
                s1.show();
        cout<<"Enter amount to withdraw:"<<endl;
        cin>>n; s1.withdrawal(n);
                cout<<"Withdrawal: Rs."<<n<<"\n";       s1.show();
                cout<<"Adding interest of 0.20 to current balance:"
                <<endl;
                s1.interest();  s1.show();
    }
    return 0;
}
```

Output:

```
mouri@intell0-hub: ~/c++-files/1
mouri@intell0-hub ~/c++-files/11-23 ./a.out
Enter type of account(s for savings, c for current):
s
Enter balance:
12000
Enter amount to deposit:
2000
Deposited: Rs.2000
Current balance:14000.00
Enter amount to withdraw:
20000
Insufficient balance.
Withdrawal: Rs.20000
Current balance:14000.00
Enter amount to withdraw:
5000
Withdrawal: Rs.5000
Current balance:9000.00
Adding interest of 0.20 to current balance:
Current balance:10800.00
```

```
mouri@intell0-hub: ~/c++-files/1
mouri@intell0-hub ~/c++-files/11-23 ./a.out
Enter type of account(s for savings, c for current):
c
Enter balance:
12000
Enter minimum balance:
2000
Enter amount to deposit:
2000
Do you want to use cheque or cash?(c for cheque)..
c
Deposited by cheque...
Deposited: Rs.2000
Current balance=14000.00
Enter amount to withdraw:
13000
Withdrawal: Rs.13000
Current balance=1000.00
Enter amount to withdraw:
2000
Withdrawal: Rs.2000
Current balance=900.00
```

.

6. Rewrite the above program using constructors.

Query:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
using namespace std;
class Account{
    string customer_name,acct_type;
    ll acct;
    public:
    Account(){}
    Account(string a,string b,ll c){
        customer_name=a;
        acct_type=b;
        acct=c;
    }
};
class Cur_acct:public Account{
    double balance,minBal;
    public:
        Cur_acct(double a,double b){
            balance=a;
            minBal=b;
        }
        void withdrawal(int n){
            if(balance<minBal)balance-=(balance*0.10);
            else if(n>balance)printf("Insufficient balance.\n");
            else balance-=n;
        }
        void deposit(int n){    balance+=n;     }
        void cheque(){
            cout<<"Deposited by cheque..."<<endl;
        }
        void show(){
            printf("Current balance=%.2f\n",balance);
        }
};
class Sav_acct:public Account
{
    double balance;
    public:
        Sav_acct(int a){    balance=a;  }
        void withdrawal(int n){
            if(n>balance)printf("Insufficient balance.\n");
            else balance-=n;
        }
        void deposit(int n){    balance+=n;     }
        void interest(){    balance+=(balance*0.20);    }
        void show(){
            printf("Current balance:%.2f\n",balance);
        }
};
int main(){
    cout<<"Enter type of account(s for savings, c for current):"<<endl;
    char c;    cin>>c;
    cout<<"Enter balance:"<<endl;
    int bal;    cin>>bal;
    if(c=='c'){
```

```
        cout<<"Enter minimum balance:"<<endl;
        int minbal;      cin>>minbal;
        Cur_acct c1(bal,minbal);
        cout<<"Enter amount to deposit:"<<endl;
        int n;  cin>>n;
        cout<<"Do you want to use cheque or cash?(c for cheque).."<<endl;
        char c; cin>>c;
        if(c=='c')c1.cheque();
        c1.deposit(n);
        cout<<"Deposited: Rs."<<n<<"\n";         c1.show();
        cout<<"Enter amount to withdraw:"<<endl;
        cin>>n; c1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";
        c1.show();
        cout<<"Enter amount to withdraw:"<<endl;         cin>>n;
        c1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";        c1.show();
    }
    else{
        Sav_acct s1(bal);
        cout<<"Enter amount to deposit:"<<endl;
        int n;  cin>>n;
        s1.deposit(n);
        cout<<"Deposited: Rs."<<n<<"\n";         s1.show();
        cout<<"Enter amount to withdraw:"<<endl;
    cin>>n;      s1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";
        s1.show();
        cout<<"Enter amount to withdraw:"<<endl;         cin>>n;
        s1.withdrawal(n);
        cout<<"Withdrawal: Rs."<<n<<"\n";        s1.show();
        cout<<"Adding interest of 0.20 to current balance:"<<endl;
        s1.interest();  s1.show();
    }
    return 0;
}
```

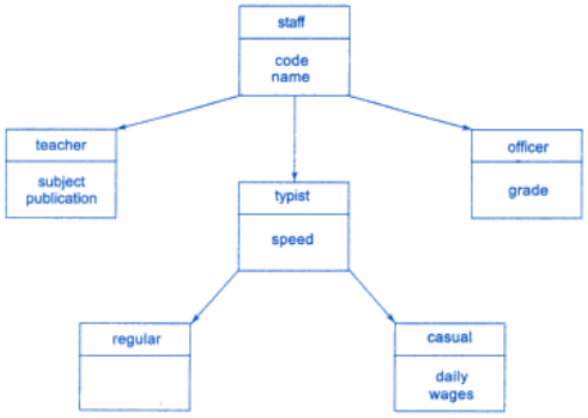
Output:

```
mouri@intell0-hub: ~/c++-files/1
mouri@intell0-hub ~/c++-files/11-23 ./a.out
Enter type of account(s for savings, c for current):
s
Enter balance:
12000
Enter amount to deposit:
2000
Deposited: Rs.2000
Current balance:14000.00
Enter amount to withdraw:
20000
Insufficient balance.
Withdrawal: Rs.20000
Current balance:14000.00
Enter amount to withdraw:
5000
Withdrawal: Rs.5000
Current balance:9000.00
Adding interest of 0.20 to current balance:
Current balance:10800.00
```

```
mouri@intell0-hub: ~/c++-files/1
mouri@intell0-hub ~/c++-files/11-23 ./a.out
Enter type of account(s for savings, c for current):
c
Enter balance:
12000
Enter minimum balance:
2000
Enter amount to deposit:
2000
Do you want to use cheque or cash?(c for cheque)..
c
Deposited by cheque...
Deposited: Rs.2000
Current balance=14000.00
Enter amount to withdraw:
13000
Withdrawal: Rs.13000
Current balance=1000.00
Enter amount to withdraw:
2000
Withdrawal: Rs.2000
Current balance=900.00
```

.

7. An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in figure below. The figure also shows the minimum information required for each class. Specify all the classes and define functions to create the database and retrieve individual information as and when required.



Program:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#define ll long long int
using namespace std;
class Staff{
    string code,name;
    public:
    Staff(){}
    void init(string a,string b){  code=a;  name=b;  }
    void getS(){
        cout<<"Code= "<<code<<endl;
        cout<<"Name= "<<name<<endl;
    }
};
class Teacher:public Staff{
    string subj,publ;
    public:
    Teacher(){}
    void init1(string c,string d,string a,string b){
                init(c,d);  subj=a;  publ=b;
    }
    void getT(){
        getS(); cout<<"Subject= "<<subj<<endl;
        cout<<"Publication= "<<publ<<endl;
    }
};
class Typist:public Staff{
    double speed;
    public:
    Typist(){}
    void init1(string a,string b,double d){
                init(a,b);      speed=d;
    }
    void getT2(){
        getS();
```

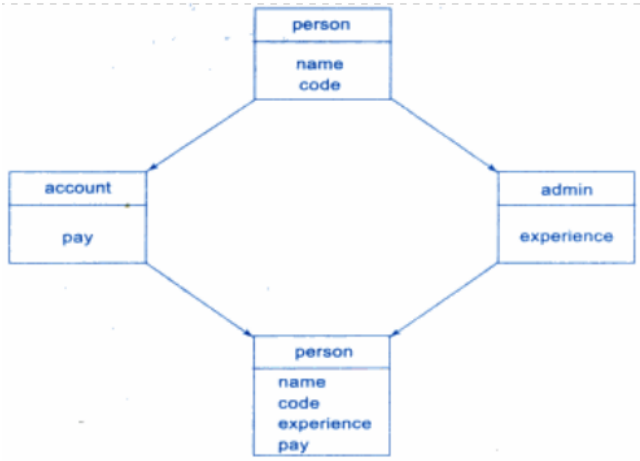
```
        cout<<"Speed= "<<speed<<" Words per minute."<<endl;
    }
};
class Officer:public Staff{
    string grade;
    public:
    Officer(){}
    void init1(string b,string c,string a){
                init(b,c);      grade=a;
    }
    void getO(){
                getS(); cout<<"Grade= "<<grade<<"\n";
    }
};
class Regular:public Typist{
    public:
    Regular(){}
    void init2(string a,string b,int s){  init1(a,b,s);  }
    void getR(){        getT2();    }
};
class Casual:public Typist
{
    int wages;
    public:
    Casual(){}
    void init2(string a,string b,int s,int t)
    {
        init1(a,b,s);
        wages=t;
    }
    void getC()
    {
        getT2();
        cout<<"Daily Wages= "<<wages<<endl;
    }
};
int main()
{
    Teacher t;
    cout<<"Staff: Teacher"<<endl;
    t.init1("0001","A","Maths","Srijan");
    t.getT();
    Officer o;
    cout<<"Staff: Officer"<<endl;
    o.init1("1001","B","A");
    o.getO();
    cout<<"Staff: Typist(Regular)"<<endl;
    Regular r;
    r.init2("2001","C",30);
    r.getR();
    cout<<"Staff: Typist(Casual)"<<endl;
    Casual c;
    c.init2("2002","D",30,10000);
    c.getC();
    return 0;
}
```

Output:

```
mouri@intell0-hub: ~/c++-files/11-23 83x45
mouri@intell0-hub ~/c++-files/11-23 ./a.out
Staff: Teacher
Code= 0001
Name= A
Subject= Maths
Publication= Srijan
Staff: Officer
Code= 1001
Name= B
Grade= A
Staff: Typist(Regular)
Code= 2001
Name= C
Speed= 30 Words per minute.
Staff: Typist(Casual)
Code= 2002
Name= D
Speed= 30 Words per minute.
Daily Wages= 10000
```

.

8. Consider a class network as shown below. Define all four classes and write a program to create, update and display the information contained in master objects.



Query:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
class Person{
    public:
    string name,code;
    void init(string a,string b){
        name=a; code=b;
    }
    virtual void display()=0;
};
class Account:virtual public Person{
    public:
    int pay;
    void init1(string a,string b,int c){
```

```
        init(a,b);      pay=c;
    }
    void display(){
        cout<<"Name= "<<name<<endl;     cout<<"Code= "<<code<<endl;
        cout<<"Pay= "<<pay<<endl;
    }
};
class Admin:virtual public Person{
    public:
    string exp;
    void init2(string a,string b,string c){
        init(a,b);      exp=c;
    }
    void display(){
        cout<<"Name= "<<name<<endl;     cout<<"Code= "<<code<<endl;
        cout<<"Experience= "<<exp<<endl;
    }
};
class Master:public Account,public Admin{
    public:
    void init3(string a,string b,string c,int d){
        init1(a,b,d);   init2(a,b,c);
    }
    void display(){
        cout<<"Name= "<<name<<endl;     cout<<"Code= "<<code<<endl;
        cout<<"Experience= "<<exp<<endl;  cout<<"Pay= "<<pay<<endl;
    }
};
int main(){
    Master p;
    p.init3("A","0001","2 years",10000);    p.display();
    return 0;
}
```

Output:

```
mouri@intell0-hub ~/c++-files/11-23 ./a.out
Name= A
Code= 0001
Experience= 2 years
Pay= 10000
```

.

9. Write a program demonstrating overriding using 'virtual'.

Query:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
class Base{
    public:
        void display(){    cout<<"Display Base"<<endl;  }
        virtual void show(){    cout<<"Show Base"<<endl;        }
};
class Derived:public Base{
    public:
```

```
        void display(){    cout<<"Display Derived"<<endl;       }
        void show(){    cout<<"Show Derived"<<endl;     }
};
int main(){
    Base b;
    Derived d;    Base *bptr;
    bptr=&b;
    bptr->display();    bptr->show();
    bptr=&d;
    bptr->display();    bptr->show();
    return 0;
}
```

Output:

```
mouri@intell0-hub ~/c++-files/11-30 ./a.out
Display Base
Show Base
Display Base
Show Derived
```

.

10. Write a program implementing Shape class, from which Triangle and Rectangle inherit. Use 'virtual'

Query:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
class Shape{
    public:
    double l,b;
    void getData(double _l,double _b){
        l=_l;   b=_b;
    }
    virtual void display(){
        cout<<"Area ="<<(l*b)<<"\n";
    }
};
class Triangle:public Shape{
    public:
    void display(){
        cout<<"Area of triangle="<<(0.5*b*l)<<"\n";
    }
};
class Rectangle:public Shape{
    public:
    void display(){
                cout<<"Area of triangle="<<(l*b)<<"\n";
    }
};
class Circle:public Shape{
    public:
    void getData(int _l,int _b=0){
        l=_l;   b=_b;
    }
```

```
    void display(){
        cout<<"Area of circle="<<(3.1412*l*l)<<"\n";
        }
};
int main()
{
    Triangle t; Rectangle r;
    Circle c;
    double l,b;
    t.getData(4,5);      t.display();
    r.getData(4,5);      r.display();
    c.getData(5);        c.display();
    return 0;
}
```

Output:

```
mouri@intell0-hub ~/c++-files/11-30 ./a.out
Area of triangle=10
Area of rectangle=20
Area of circle=78.53
```

.

11. Rewrite the previous program without using 'virtual'.

Query:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
class Shape{
    public:
    double l,b;
    void getData(double _l,double _b){
        l=_l;   b=_b;
    }
    void display(){
        cout<<"Area ="<<(l*b)<<"\n";
    }
};
class Triangle:public Shape{
    public:
    void display(){
        cout<<"Area of triangle="<<(0.5*b*l)<<"\n";
    }
};
class Rectangle:public Shape{
    public:
    void display(){
        cout<<"Area of rectangle="<<(l*b)<<"\n";
    }
};
class Circle:public Shape{
    public:
    void getData(int _l,int _b=0){
        l=_l;   b=_b;
    }
```

```
};
int main()
{
    Triangle t; Rectangle r;
    Circle c;
    t.getData(4,5);      t.display();
    r.getData(4,5);      r.display();
    c.getData(5);        c.display();
    return 0;
}
```

Output:

```
mouri@intell0-hub ~/c++-files/11-30 ./a.out
Area of triangle=10
Area of rectangle=20
Area =0
```

.

12. Write a function template for finding the minimum value contained in an array.

Query:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
template<class R>
R minimum(R a[],int n){
    R m;     m=a[0];
    for(int i=1;i<3;i++)if(m>a[i])m=a[i];
    return m;
}
int main(){
    int x[3]={10,21,3};
    cout<<minimum(x,3)<<"\n";
    return 0;
}
```

Output:

```
mouri@intell0-hub ~/c++-files/12-7 ./a.out
3
```

.

13. Write a program containing a possible exception. Perform exception handling with multiple catch.

Query:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
void excp(){
    int x=4;    int y=4;
    if(x==y)throw(x-y);
    else cout<<"Its fine"<<endl;
}
int main(){
    try{        excp();    }
    catch(int i){
        cout<<"Both are equal."<<endl;
    }
    catch(char c){
        cout<<"Character is found."<<endl;
    }
    return 0;
}
```

Output:

```
mouri@intell0-hub ~/c++-files/12-7 ./a.out
Both are equal.
```

.

14. Write a program to demonstrate the concept of rethrowing an exception.

Query:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
void excp1(){
    int x=9;    int y=7;
    if(x-y==2)throw('A');
    else cout<<"Its fine."<<endl;
}
int main(){
    try{        excp1();    }
    catch(char c){
        if(c=='B')cout<<"The diff is 2."<<endl;
        else throw;
    }
    return 0;
}
```

Output:

```
mouri@intell0-hub ~/c++-files/12-7 ./a.out
terminate called after throwing an instance of 'char'
[5]    12111 abort (core dumped)  ./a.out
```

.

15. Write a class template to represent a generic vector. Include member functions to perform the following taks:
∗ to create a vector
∗ to modify the value of a given element
∗ to multiply by a scalar.

Query:

```
#include<iostream>
#include<cstdio>
#include<string>
#include<stdlib.h>
#include<cmath>
#define ll long long int
using namespace std;
template<class T>
class vector
{
    T *v;
    public:
        void init(int size){
            v=new int[size];
            cout<<"Enter values"<<endl;
            for(int i=0;i<size;i++)cin>>v[i];
        }
        T multiply(vector &a,vector &b){
            T sum=0;
            for(int i=0;i<3;i++)sum+=a.v[i]*b.v[i];
            return sum;
        }
};
int main(){
    vector<int> v1;      vector<int> v2;
    vector<int> v;
    cout<<"Enter 3 values for v1:"<<endl;        v1.init(3);
    cout<<"Enter 3 values for v2:"<<endl;        v2.init(3);
    cout<<"Product="<<v.multiply(v1,v2)<<"\n";
    return 0;
}
```

Output:

```
mouri@intell0-hub ~/c++-files/12-7 ./a.out
Enter 3 values for v1:
Enter values
1 2 3
Enter 3 values for v2:
Enter values
4 5 6
Product=32
```

.