1. Define a class Employee with the following specification :

Private :

•Ename and Deptname of char ,

•Salary and bonus of type float.

•Callbonus() function to calculate bonus given to an employee according to the following conditions

**Deptname bonus**

Accounts 4%of salary

HR 5% of salary

IT 3% of salary

Marketing 4% of salary

For all others 2% of salary

Public members:

• Constructor to initialize ename and deptname to NULL and salary and bonus to 0.

• A function read() to get from the user the details of employee and call function callbonus().

Class employee

{

Char ename[30], deptname[30];

Float salary, bonus;

Void Callbonus()

{

If(strcmp(deptname, “accounts”)==0)

Bonus= 4\*salary/100;

Else if(strcmp(deptname, “HR”)==0)

Bonus= 5\*salary /100;

Else if (strcmp(deptname, “IT”)==0)

Bonus= 3\*salary /100;

Else if(strcmp(deptname, “MArketing”)==0)

Bonus= 4\*salary /100;

Else

Bonus = 2\*salary/100;

}

Public:

Employee()

{

Strcpy(name, “NULL”);

Strcpy(deptname , “NULL”);

Bonus=0; salary=0;

}

Void read()

{

Cout<<”enter empname, deptname, salary”;

Cin>>empname>>deptname>>salary;

Callbonus();

}

Void display()

{

Cout<<empname <<deptname<<salary<<bonus;

}

};

Void main()

{

Employee e;

e.read();

e.display();

}

2. Define a class competition with complete function definitions in C++ with following specification.

Private:

• Event\_No of type integer

• Description of type string, size[30]

• Score of type integer

• Qualified of type character

Public :

A constructor functions to assign initial values Event\_no as 101 description as “ state level”, score as 50 and qualified as ‘N’

• Input( ) To take input for Event\_No, description and score.

• Award ( ) To award qualified as ‘y’ if score is more than the cut off score passed as argument to the function else ‘N’

• Show( ) To display all the details

Write a function main ( ) accepting the input for cutoffscore and call the necessary functions.

class competition

{

Private:

Int Event\_No ;

Char Description[30];

Int Score;

Char Qualified;

Public :

Competition(int e =40, char d[30]= “state level”, int s =50)

{

Event\_no=e;

Strcpy(description, d);

score = s ;

qualified = ‘N’;

}

// Competition c(30, “district level”, 70);

Competition c( 40)

void Input( )

{

Cin>> Event\_No >>description>> score;

}

Void Award (int cutoff )

{

If( score > cutoff)

qualified = ‘y’

else

qualified = ‘N’;

}

Void Show( )

{

cout<< event\_no <<description << score<<qualified;

}

};

Void main()

{

Competition c;

Int cs;

c.input();

cout<<”enter the cutoff score”;

cin>>cs;

c.award(cs)

c.show();

}

3. Define a class Tour with complete function definitions with following specification.

Private members:

Tcode Integer, adults, children and distance , total.

Public functions :

AssignFare() which will calculate and assign the value to Total based on the following condition

For Adults : fare is Rs. 500 if Distance is >=1500 . And fare get reduced by 25% if distance is <1500.

For children : fixed rate is Rs. 50. Find total fare.

Also write functions getdata() and display () to enter and display the values of private variables.

Class tour

{

Int tcode, adults, children, distance, total, fare;

Public:

Void assignfare()

{

If(distance >=1500)

Fare = 500;

Else

Fare = 500- (25\*500/100);

Total = adults\* fare +children\*50;

}

Void getdata()

{

Cout<<”enter the value “;

Cin>>tcode>>adults>>children>>distance;

}

Void display()

{

Cout<<”total fare is “<<total;

}

};

Void main()

{

Tour t;

t.getdata();

t.assignfare();

t.display();

}

4. Define a class ‘‘Library’’ in C++ with the following specifications :

**Private :**

• code — type integer

• name — character array of size 20

• game\_name — character array of size 20

• mtype — type character (P for permanent member and Tfor temporary member)

• fees — type float

**Public :**

• A parameterised constructor to assign the values for data

members code, name, game\_name and mtype.

• Calculate()-to calculate fees according to the following criteria.

Fees for different games.

**Game-name Fees for Permanent Member**

Cricket 1000

Tennis 2000

Badminton 3000

If the member is temporary then the fees is double of permanentmember.

• Display()-to display all the data members on the screen.

Write a function main() to call necessary functions.

Class library

{

Int code; float fees;

Char name [30], gname[20], mtype;

**Public:**

**Library(int c, char n[30], char gn[20], char mt)**

**{**

**Code= c;**

**Strcpy(name, n);**

**Strcpy(gname, gn);**

**Mtype = mt;**

**}**

**Void calculate()**

**{**

**If(mtype ==’P’)**

**{**

**If (strcmp(gname, “Cricket”) ==0)**

**Fees= 1000;**

**Else if(strcmp(gname, “Tennis”)==0)**

**Fees = 2000;**

**Else**

**If(strcmp(gname, “Badminton”)==0)**

**Fees = 3000;**

**}**

**Else**

**{**

**If (strcmp(gname, “Cricket”) ==0)**

**Fees= 2000;**

**Else if(strcmp(gname, “Tennis”)==0)**

**Fees = 4000;**

**Else**

**If(strcmp(gname, “Badminton”)==0)**

**Fees = 6000;**

**}**

**}**

**Void display()**

**{**

**Cout<<”the fees are “<<fees;**

**}**

**};**

**Void main()**

**{**

**Library l(1, “x”, “Cricket”, ‘P’);**

**l.calculate();**

**l.display();**

**}**

5. . Answer the questions (i), (ii) and (iii) after going through the following class.

class Match

{

int Time;

public :

Match (int y) // Constructor 1

{

Time = y;

}

Match (Match &m); // Constructor 2

};

1. Create an object of Match such that it invokes constructor 1 with time passed as 7.

Match m(7);

1. Write the function definition for constructor 2.

Match(match &m)

{

Time= m.Time;

}

Void main()

{

Match m1(7);

Match m2(m1);

1. Identify the type of constructors depicted in the above class

Constructor 1 = parameterized constructor

Constructor 2 = copy constructor

6. Answer the questions i) ii) and iii) after going through the following program

class sports

{

int time;

public :

sports ( ) //function 1

{

time = 0;

}

void display ( ) // function 2

{

cout << “ It is a sports festival”;

}

sports ( int duration ) // function 3

{

time = duration;

}

sports(sports &sf ) //function 4

{

time =sf.duration;

}

};

i) Which feature of object oriented programming is demonstrated using function 1,

function 3 and function 4 ? ---- Constructors

ii) Which category of constructor function 4 belongs to and what is the purpose of using it? ----------Copy constructor

iii) Identify the type of constructor depicted in function 3.-------- Parameterized constructor

iv) Identify the type of constructor depicted in function 1. When will it be invoked?

Default constructor and it is involved when a new object is being created.

v) size of the object

Example of copy constructor

class student

{

Int rollno, maths, science, English, total, per;

Public:

Void getdata()

{

Cin>>rollno>>maths>>science>>English;

}

Void calc()

{

Total = maths+science+English;

Per= total\*100/300;

}

Void average(student s1, student s2)

{

Maths = (s1.maths +s2.maths)/2;

Science = (s1.science +s2.science)/2;

English = (s1.english +s2.english)/2;

}

Void average1(student s1)

{

English = (English+s1.english)/2;

Maths =( maths +s1.maths)/2;

science =( science +s1.science)/2;

}

Student(student &s)

{

Rollno= s.rollno;

Strcpy(name, s.name);

Maths= s.maths;

English= s.english;

Science= s.science;

}

Void display()

{

Cout<<total<<per;

}

};

Void main()

{

Student s1, s2;

S1.getdata();

S2.getdata();

Student S3(s1);

S3.calculate();

S3.display();

Student s4;

S4.average(s1,s2);

S4.calculate();

S4.display();

S2.average1(s1);

S2.calculate();

S2.display();

}

Inheritance

Single level A- > B

Multi level A -> B -> C;

Multiple (A & B )-> C

Hierarchical – A- > B & C

Hybrid -> hierarchical and multiple

Single inheritance :

Class student

{

Protected :

Int rollno;

Char name[20];

Public :

Void getdata() { cin>>rollno>>name;}

Void display() { cout<<rollno <<name; }

};

Class marks : public student

{

float english, cs, chem;

Float total, per;

Public :

Void getmarks() { cin>>English>>cs>>chem; }

Void dispmarks() {

cout <<English <<cs <<chem ; }

Void calculate ()

{

Total = chem+English+cs;

Per = totat\*100/300;

};

Void main()

{

Marks m;

m.getdata();

m.getmarks();

m.calculate();

m.display();

m.dispmarks();

}