Programs (15 marks)

1.a)Write a C++ program to find average of 3 integer numhers and average of 3 float numbers.(Use function overloading)

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

int add(int,int,int);
float add(float,float,float);

int main()
{
int a1,a2,a3;
float f1,f2,f3;

cout<<"Enter Integer Details:";
cin>>a1>>a2>>a3;

cout<<"Enter Float Details:";
cin>>f1>>f2 >>f3;

cout<<"\n Avarage of Integer "<<add(a1,a2,a3);
cout<<"\n Avarage of float "<<add(f1,f2,f3);

return 0;
}

int add(int a1,int a2,int a3)
{
return ((a1+a2+a3)/3);
}
float add(float  f1,float  f2,float f3)
{
return float((f1+f2+f3)/3);
}
```

2.a) a.Write C++ program to check maximum and minimum of two integer numbers (Use inline function and conditional operators)

```
#include<iostream>
#include<conio.h>
using namespace std;
class max1
{
public:

inline int maximum(int a,int b)
{
```

```
return a>b?a:b;
}
  inline int minimum(int a,int b)
{
return a<b?a:b;
}
};
int main()
{
int a,b;
max1 m;
cout<<"\n Enter two number:";
cin>>a>>b;

cout<<"\nNumber 1st: " <<a<<endl;
cout<<"\nNumber 2nd: "<<b<<endl;
 cout<<"\nMaximum number is: "<<m.maximum(a,b)<<endl;
 cout<<"\nMinimum is: "<<m.minimum(a,b)<<endl;
}
```

3.a.)Write a C++ program to calculate Volume of cone,sphere and cylinder by   using function overloading.

```
 #include<iostream>
using namespace std;
float vol(int,int);//cylinder
float vol(float);//sphere
int vol(float,int);//cone

int main()
{
int radius,height,radius2,height2;
float radius1;

cout<<"Enter radius and height of a cylinder:";
cin>>radius>>height;

cout<<"Enter radius of sphere: ";
cin>>radius1;

cout<<"Enter radius and height of a cone:";
cin>>radius2>>height2;

cout<<"\n Volume of cylinder is "<<vol(radius,height);//cylinder
cout<<"\n Volume of sphere is "<<vol(radius1);//sphere
cout<<"\n Volume of cone is "<<vol(radius2,height2);//cone

return 0;
}
```

```
float vol(int radius,int height)//cylinder
{
return(3.14*radius*radius*height);
}

float vol(float radius1)//sphere
{
return((4*3.14*radius1*radius1*radius1)/3);
}

int vol(float radius2,int height2)//cone
{
return(0.33*3.14*radius2*radius2*height2);
}
```

---

4. a) Write a C++ program to create a class which contains two data members. Write member functions to accept display and swap two entered numbers using call by reference.

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

void swap(int &x,int &y)
{
        int temp;
        temp = x;
        x = y;
        y = temp;
        return;
}
int main()
{
        int a = 100;
        int b = 200;
    cout<<"Before swap ,value of a :" <<a<<endl;
    cout<<"Before swap ,value of b :" <<b<<endl;
    swap(a,b);
    cout<<"After swap ,value of a :" <<a<<endl;
    cout<<"After swap ,value of b :" <<b<<endl;
    return 0;
}
```

---

5. a)Write a C++ program to create a class Worker with data members as Worker_Name, No_of_Hours_worked, Pay_Rate. Write necessary member

functions to calculate and display the salary of worker. (Use default value for Pay_Rate)

```
#include<iostream>
using namespace std;
class worker
{
    char name[10];
    int hr;
    public:
    void accept()
    {
       cout<<"enter name";
       cin>>name;
       cout<<"enter hours";
        cin>>hr;
     }
     void calculate(int rate=20)
     {
        cout<<"salary of worker is Rs."<<(hr*10)*rate;
      }
    };

int main()
{
   worker w;
   w.accept();
   w.calculate();

}
```

---

6. a) /*Write a C++ program to create two Classes Square and Rectangle. Compare area of both the shapes using friend function. Accept appropriate data members for both the classes .*/

```
#include<iostream>
using namespace std;
#include<conio.h>
class Square
{
friend class Rectangle;      // declaring Rectangle as friend class
int side;
public:
Square ( int s )
{
side = s;
}
};

class Rectangle
{
int length;
int breadth;
public:
int getArea()
{
return length * breadth;
}
void shape( Square a )
{
length = a.side;
breadth = a.side;
}
};

int main()
{
Square square(5);
Rectangle rectangle;
rectangle.shape(square);
cout << " Area of rectangle is   "<<rectangle.getArea();
return 0;
}
```

7 a) /*Write a C++ program to create a class Number, which contain static data member ' cnt' and member function ' Display() '. Display() should print number of times display operation is performed irrespective of the object responsible for calling Display().*/

```
#include<iostream>
using namespace std;
class Number
{
static int cnt;
public:
static void display()
{
cout<<"\n Number of time call show: "<<cnt;
cnt++;
}
};
int Number::cnt;
int main()
{
Number s1,s2,s3,s4;
s1.display();
s2.display();
s3.display();
s4.display();
return 0;
}
```

```
8 a) /*Consider the following C++ class
class Person
{
char Name [20];
char Add r [30];
float Salary;
float  tax_amount;
public:
II member functions
};
Calculate tax amount by checking salary of a person
•For salary <=20000 tax rate=0
•For salary >20000 ||< =40000  tax rate= 5% of salary.
•For salary >40000  tax rate =10% of salary.*/

#include<iostream>
using namespace std;
class person
{
char name[20];
char addr[20];
float sal,tax;
 public:
void get()
{
cout<<"Enter the name, address, salary : \n";
cin>>name>>addr>>sal;
}
void put()
{
cout<<"Person Information:\n";
cout<<"Name\tAddress\tSalary\tTax: \n";
cout<<"======================================================\n"
;
cout<<name<<"\t"<<addr<<"\t"<<sal<<"\t"<<tax<<endl;
}
void cal_tax()
{
if(sal<=20000) //salary <=20000
{
tax=0;
}
else if(sal>=20000||sal<=40000)//salary >20000 11< =40000    tax rate= 5%
of salary.
{
tax=(sal*5)/100;
}
else if(sal >40000) //salary >40000  tax rate =10% of salary
{
tax=(sal*10)/100;
}
}
};
int main()
{
person p;
p.get();
p.cal_tax();
p.put();
}
```

```
9 a) /*Write a C++ program to create a class Account with data members
Acc_number, Acc_type and Balance. Write member functions to accept and
display ' n' account details. (Use dynamic memory allocation)*/
#include<iostream>
#include<conio.h>
#include<stdlib.h>
using namespace std;
 class  Account
```

```
{
public:
int Acc_no,Balance;
char Acc_type[30];

public:
Account()  {  cout << "Constructor" << endl; }
~Account() { cout << "Destructor" << endl; }
void get_data()
{
cout<<"\n Enter Acc_no.:";
cin>>Acc_no;
cout<<"\n Enter Acc_type :";
cin>>Acc_type;
cout<<"\n Enter Balance :";
cin>>Balance;
}
void display_data()
{
cout<<"\t"<<Acc_no<<"\t"<<"\t"<<Acc_type<<"\t"<<Balance;
}
};
int main()
{

 int num,i;
 Account* a = new Account[4];
 delete [] a; // Delete array
 cout<<"\n How many records u want?: ";
cin>>num;
 for(int i=0;i<num;i++)
{
a[i].get_data();
}
for(i=0;i<num;i++)
{
a[i].display_data();
}
return 0;
 }
```

10.a)Write a C++ program to create a class Date with data members day, month, and year. Use default. and parameterized constructor to initialize date and display date in dd-Mon-yyyy format. (Example: Input: 04-01-2021 Output: 04-Jan-2021)

```
#include<iostream>
using namespace std;
class date
{
int dd,mm,yyyy;

public:
date(){}//default constructor
date(int d,int m,int y) // creating parameterized constructor
{
dd=d;
mm=m;
yyyy=y;
}
void disp()
{
if(mm>12)
cout<<"Invalid Month."; //mm is month if it is greater than 12 then it is
invalid.
else
cout<<"Input   : "<<dd<<"/"<<mm<<"/"<<yyyy<<endl;
if(mm==1)
{
cout<<"Output  : "<<dd<<"/"<<"jan"<<"/"<<yyyy;
}
else if(mm==2)
{
cout<<"Output  : "<<dd<<"/"<<"Feb"<<"/"<<yyyy;
}
else if(mm==3)
{
cout<<"Output  : "<<dd<<"/"<<"mar"<<"/"<<yyyy;
}
else if(mm==4)
{
cout<<"Output  : "<<dd<<"/"<<"apr"<<"/"<<yyyy;
}
else if(mm==5)
{
cout<<"Output  : "<<dd<<"/"<<"may"<<"/"<<yyyy;
}
else if(mm==6)
{
cout<<"Output  : "<<dd<<"/"<<"jun"<<"/"<<yyyy;
}
else if(mm==7)
 {
 cout<<"Output  : "<<dd<<"/"<<"july"<<"/"<<yyyy;
 }
else if(mm==8)
{
cout<<"Output  : "<<dd<<"/"<<"Aug"<<"/"<<yyyy;
}
else if(mm==9)
{
cout<<"Output  : "<<dd<<"/"<<"sep"<<"/"<<yyyy;
}
else if(mm==10)
{
cout<<"Output  : "<<dd<<"/"<<"oct"<<"/"<<yyyy;
}
else if(mm==11)
{
cout<<"Output  : "<<dd<<"/"<<"Nov"<<"/"<<yyyy;
}
else if(mm==12)
{
cout<<"Output  : "<<dd<<"/"<<"Dec"<<"/"<<yyyy;
}
}
};
int main()
{
int d,m,y;
cout<<"Enter a date : ";
cin>>d;
cout<<"Enter a month: ";
cin>>m;
cout<<"Enter a year : ";
cin>>y;
date d1(d,m,y);
d1.disp();
```

```
return(0);
}
```

11 a) Write a C++ program to create a class
Product with data members Product_id, Product_Name, Qty, Price. Write member functions to accept and display Product information also display number of objects created for Product class. (Use Static data member and Static member function)

```
#include<iostream>
using namespace std;
class Product
{
int id,price,Qty;
char i_name[20];
static int cnt;
public:
void getdata()
{
cout<<"\n Enter Product Id\t\t";
cin>>id;
cout<<"\n Enter Product name\t\t";
cin>>i_name;
cout<<"\n Enter Product price\t";
cin>>price;
cout<<"\n Enter Product Qty\t";
cin>>Qty;
cnt++;
}
void display()
{
//cout<<"\n********************OUTPUT**********************";
cout<<"\n\nProduct code =\t"<<id;
cout<<"\nProduct name =\t"<<i_name;
cout<<"\nProduct price =\t"<<price;
cout<<"\nProduct QTY =\t"<<Qty;
}
static void nob()
{
cout<<"\n number of objects created for class are\t"<<cnt;
}
};
```

```
int Product::cnt;
int main()
{
Product ob[10];
int n,i;
cout<<"\n Enter how many items : ";
cin>>n;
for(int i=0;i<n;i++)
ob[i].getdata();
for(i=0;i<n;i++)
ob[i].display();
ob[n-1].nob();
return 0;
}
```

12 A) Write C++ program to accept radius of circle . Calculate and display diameter, circumference and area of circle.

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

 inline float dimeter(float r)
{
return(r *2);
}
inline float circlearea(float r)
{
return(3.14*r*r);
}
inline float circumference(float r)
{
return(3.14*2*r);
}
int main()
{
float radius;

 cout<<"\n\nEnter the radius of the circle:";
cin>>radius;
cout<<"\nDiameter of Circle:"<< dimeter(radius);
cout<<"\nArea of Circle:"<< circlearea (radius);
cout<<"\nCircumference of Circle:"<< circumference(radius);
return 0;
 }
```

13) a.Write a C++ program to create a class Machine with data members Machine_Id, Machine_Name, Price. Create and initialize all values of Machine object by using parameterized constructor and copy constructor. Display details of Machine using setw() and setprecsiion().

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

class Machine
{
private:
int Machine_Id, Price;   //data members
char name[10];
public:
Machine(int x1, int y1,char *z1)
{
strcpy(name,z1);
Machine_Id = x1;
Price = y1;
 }
 /* Copy constructor */
Machine (const Machine &sam)
{
strcpy(name,sam.name);
Machine_Id=sam.Machine_Id;
Price=sam.Price;
 }
 void display()
{
cout<<"\n Name:\t"<<name;
cout<<"\n MachineID:\t"<<setprecision(2)<<Machine_Id;
cout<<"\n Price:\t"<<setw(3)<<Price;
 }
};
/* main function */
int main()
```

```
{
Machine obj1(123, 1534,"archana");     // Normal constructor
Machine obj2 = obj1;      // Copy constructor
cout<<"Normal constructor : ";
obj1.display();
cout<<"Copy constructor : ";
obj2.display();
return 0;
}
```

---

14 a) Write a C++ program to create a class Distance with data members meter and centimeter to represent distance . Write a function Larger( ) to return the larger of two distances. (Use this pointer)

```
#include<iostream>
using namespace std;
class dist
{
int meter;
float centimeter;
public :
void get()
{
cout<<"Enter meter value";
cin>>meter;
cout<<"Enter centimeter";
cin>>centimeter;
}
void put()
{
cout<<"\t meter="<<this->meter;
cout<<"\t centimeter="<<this->centimeter;
}
dist larger(dist d1,dist d2)
{
if(d1.meter>d2.meter)
return d1;
else if(d1.meter<d2.meter)
return d2;
else
{
if(d1.centimeter>d2.centimeter)
return d1;
else
return d2;
}
}
};
int main()
{
dist d1,d2,d3;

cout<<"Enter First distance\n";
d1.get();
cout<<"Enter second distance\n";
d2.get();
d3=d3.larger(d1,d2);
cout<<"\nLarger Distance\n";
d3.put();

return 0;
}
```

---

25 marks programs

1) b) **Create a class ComplexNumber containing members as:**
- real
- imaginary

Write a C++ program to calculate and display the sum of two complex numbers. (Use friend function and return by object)

```
#include<iostream>
using namespace std;
#include<conio.h>
class complex
{
 float real;
 float imaginary;
 public:
complex(float a=0.0,float b=0.0)
{
  real=a;
 imaginary=b;
}
 void display()
{
 cout<<"\n real number :"<<real;
 cout<<"\n imaginary number:"<<imaginary<<"i";
 }
friend complex operator +(complex,complex);
};

complex operator +(complex c1,complex c2)
{
complex temp;
temp.real=c1.real+c2.real;
temp.imaginary=c1.imaginary+c2.imaginary;
return temp;
}
int main()
{
float n1,n2,n3,n4;

cout<<"\n Enter 1st real and imaginary no :- ";
 cin>>n1;
 cin>>n2;
 complex c1(n1,n2);

 cout<<"\n Enter 2nd real and imaginary no :- ";
 cin>>n3;
 cin>>n4;
 complex c2(n3,n4);

cout<<"\n 1st complex no is :  ";
c1.display();
cout<<"\n 2nd complex no is :  ";
 c2.display();

 complex c3;
 c3=c1+c2;
 cout<<"\n After adding of float and imaginary no are :";
 c3.display();
return 0;
}
```

1 Create a class for inventory of Mobile containing Model_Number, Company_Name, Price, Stock as data members. Mobile can be sold, if stock is available, otherwise purchase will be made. Write a C++ program to perform following functions :

i. To accept and display mobile details
ii. To sale a Mobile (Sale contains Model_Number & Quantity of mobile)
iii. To Purchase a Mobile (Purchase contains Model_Numbet & Quantity of mobile)

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

```
class inventory
{
int modelnum,price,stock;
char company[10];
public:
void getput()
{
cout<<"\n\nEnter model number:\t";
cin>>modelnum;
cout<<"\nEnter mobile company name:\t";
cin>>company;
cout<<"\nEnter price:\t";
cin>>price;
cout<<"\nEnter stock:\t";
cin>>stock;
cout<<"\n\n1.Model number\t"<<modelnum;
cout<<"\n2.Company name\t\t"<<company;
cout<<"\n3.Price\t"<<price;
cout<<"\n4.Stock\t"<<stock;
}
void sale()
{
int nosale;
cout<<"\n\nModel number\t"<<modelnum;
cout<<"\nStock\t"<<stock;
cout<<"\nEnter quantity to sale\t";
cin>>nosale;
stock=stock-nosale;
cout<<"\nStock remaining\t"<<stock;
}
void purchase()
{
int nopurchase;
cout<<"\n\nModel number\t"<<modelnum;
cout<<"\nStock\t"<<stock;
cout<<"\nEnter quantity to purchase\t";
cin>>nopurchase;
stock=stock+nopurchase;
cout<<"\nStock remaining\t"<<stock;
}
};
int main()
{
int i,n,ch;
inventory in[30];
do
{
cout<<"\n\n1.Accept and display details\t";
cout<<"\n2.To sale mobile\t";
cout<<"\n3.To purchase mobile\t";
cout<<"\n4.Exit\t";
cout<<"\n\nEnter choice\t";
cin>>ch;
switch(ch)
{
case 1:
cout<<"How many records you want to enter:\t";
cin>>n;
for(i=0;i<n;i++)
in[i].getput();
break;

case 2:
for(i=0;i<n;i++)
in[i].sale();
break;
case 3:
for(i=0;i<n;i++)
in[i].purchase();
break;
case 4:
exit(0);
default:
cout<<"Wrong choice";
}
}while(ch!=4);
return 0;
}
```

```
----------------------------------------------------------------------------------------------------
```

1. b) /*Create a C++ base class Shape. Derive three different classes Circle, Sphere and Cylinder from shape class. Write a C++ program to calculate area of Circle, Sphere(4 π r2) and Cylinder(2πr (h + r)). (Use pure virtual function). */

```
#include<iostream>
using namespace std;
#include<conio.h>
#include<stdlib.h>
class shape
{
public:
virtual void area()=0;
};
```

```
class circle:public shape
{
public:
int r;
void getc()
{
cout<<"\nenter the radius:";
cin>>r;
}
void area();
};
class Sphere:public shape
{
public:
int r1;
void getr()
{
cout<<"\n Enter the radius:";
cin>>r1;
}
void area();
};
 class Cylinder:public shape
{
public:
int r2,h;
void gett()
{
cout<<"\n Enter the radius2 & height:";
cin>>r2>>h;
}
void area();
};

void circle::area()
{
cout<<(3.14*r*r);
}
void Sphere::area()
{
cout<<(4*3.14*r1*r1);
}
```

```
void Cylinder::area()
{
int d=h+r2;
cout<< d;
cout<<(2*3.14*r2*d);
}
int main()
{

int ch;
circle c;
Sphere S;
Cylinder t;

do
{
cout<<"\n1.Area of circle";
cout<<"\n2.Area of Sphere";
cout<<"\n3.Area of cylinder";
cout<<"\n4.Exit";
cout<<"\nEnter your choice:\t";
cin>>ch;

switch(ch)
{
case 1:
c.getc();
cout<<"Area of circle:";
c.area();
break;

case 2:
S.getr();
cout<<"Area of Sphere:";
S.area();
break;

case 3:
t.gett();
cout<<"Area of cylinder:";
t.area();
break;
case 4:
exit(0);
}
}while(ch!=4);
return 0;
};
```

1 B)Create a base class Student with data members Roll_No, Name.Derives two classes from it, class Theory with data members M l , M2, M3, M4 and class Practical with data members P l , P2. Class Result(Total_Marks, Perce ntage , Grade) inherit s both Theory and Practical classes.(Use concept of Virtual Base Class and protected access specifiers)

Write a C++ menu driven program to perform the following functions:
i.Accept Student Information
ii.Display Student Information
iii.Calcul ate Total_marks, Percentage and Grade.

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

class student
{
protected:
int  rno;
char name[20];
public:
void getdetails();
};
 class Theory:public virtual student
{
protected:
int mark1,mark2,mark3,mark4;
public:
void getmarks();
};
 class Practical :virtual public student
{
protected:
int p1,p2;
public:
void getpractical();
```

```
};
 class result :public Theory,public Practical
{
int total_marks;
float per;
char grade[10];
public:
void calc();
void sort(result& ,result&);
void display();
};
 void student::getdetails()
{
cout<<"\n enter roll no and name  :";
cin>>rno>>name;
}
 void Theory::getmarks()
{
cout<<"\n enter marks of four subject :";
cin >>mark1>>mark2>>mark3>>mark4;
}
 void Practical::getpractical()
{
cout<<"\n enter  Practical Details  :";
cin>>p1>>p2;
}
 void result::calc()
{
int i;
total_marks=mark1+mark2+mark3+mark4+p1+p2;
per=total_marks/(float)6;
if(per<50)
strcpy(grade,"C");
else
if(per<60)
strcpy(grade,"B");
else
if(per<75)
strcpy(grade,"A");
else
strcpy(grade,"A+");
cout<<"\n calculation complete\n";
```

```
}
 void result::sort(result &r1,result &r2)
{
result rt;
if(r1.total_marks>r2.total_marks)
{
rt=r1;
r1=r2;
r2=rt;
}
}
 void result::display()
{
cout<<"\n roll no="<<rno<<"\n name ="<<name;
cout<<"\n mark1="<<mark1<<"\n mark2 ="<<mark2<<"\n
mark3="<<mark3<<"\n mark4="<<mark4;
cout<<"\n Practical P1="<<p1<<"\n Practical P2="<<p2<<"\n percentage
="<<per<<"\n grade ="<<grade;
}
 int main()
{
int n,i,ch,j;
result r[20];

do
{
cout<<"\n MENU \n";
cout<<"\n 1.build master table \n 2. calculate total & grade \n";
cout<<"\n 3.display result in asscending order \n" ;
cout<<"\n 4. Exit \n enter your choice :  ";
cin>>ch;
switch(ch)
{
case 1:
cout<<"\n how many student :";
cin>>n;
for(i=0;i<n;i++)
{
cout<<"enter student detailse \n";
r[i].getdetails();
r[i].getmarks();
r[i].getpractical();
}
break;
 case 2:
for(i=0;i<n;i++)
r[i].calc();
break;
 case 3:
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
r[i].sort(r[i],r[j]);
r[i].display();
}
break;
 }
}while(ch<=3);
return 0;
}
```

1 B) Create a C++ class for student having following members-Rollno,Name,No. of subjects,Marks of each subject (no. of subjects varies for each student) Write a parameterized constructor which initialises Rollno,Name & no. of Subjects & creates the array of marks dynamically.Display the details of all students with percentage & class obtained.

```
#include<iostream>
 using namespace std;
class student
{
int rollno;
char name[10];
int tot_sub;
char sub_name[10][30];
int marks[10];
int tot_marks;
float per;

public:
void getdata()
{
cout<<"\n\n Enter the Roll no: ";
cin>>rollno;
cout<<"\n\n Enter the name: ";
cin>>name;
cout<<"\n\n How many subject: ";
cin>>tot_sub;
```

```
 for(int i=0;i<tot_sub;i++)
{
cout<<"\n\n Enter the subject name; ";
cin>>sub_name[i];
cout<<"\n\n Enter the marks: ";
cin>>marks[i];
}
}
public:void display()
{
tot_marks=0;
cout<<"\n Roll number:"<<rollno;
cout<<"\n Student name: "<<name;
for(int i=0;i<tot_sub;i++)
{
cout<<"\n Subject name: "<<sub_name[i];
cout<<"\n Subject mark: "<<marks[i];
tot_marks=tot_marks+marks[i];
}
per=tot_marks/tot_sub;
cout<<"\n\n Total obtain marks: "<<tot_marks;
cout<<"\n\n Percentge: "<<per;
if(per>=70)
cout<<"\n \n Grade=Dist";
else if(per>=60)
cout<<"\n \n Grade=A";
else if(per>=50)
cout<<"\n \n Grade=B";
else if(per>=40)
cout<<"\n \n Grade=Pass";
else
cout<<"\n \n Grade=Fail";
}
};
 int main()
{
student s;
s.getdata();
s.display();
return 0;
}
```

1 B) Create a C++ class VisitingStaff with data members Name, No_of_Subjects, Name_of_Subjects[],Working_hours, Total_Salary. (Number of subjects varies for a Staff).Write a parameterized constructor to initialize the data members and create an array for Name_of_Subjectsdynamically . Display Visiting Staff details by calculating salary. (Assume remuneration Rs. 300 per working hour)

```
#include<iostream>
using namespace std;
class VisitingStaff
{
char name[10];
int tot_sub;
char sub_name[10][30];
int wh,k;
public:
void getdata()
{
cout<<"\n\n Enter the name:- ";
cin>>name;
cout<<"\n\n How many subject-: ";
cin>>tot_sub;
 for(k=0;k<tot_sub;k++)
 {
cout<<"\n\n Enter the subject name;- ";
cin>>sub_name[k];
}
cout<<"\n\n Enter the Wh: ";
cin>>wh;
}

void display()

{
cout<<"\n Name of Staff : "<<name;
cout<<"\n Total Subjects :"<<tot_sub;
for(k=0;k<tot_sub;k++)
{
cout<<"\n Subject Name:"<<sub_name[k];
}
cout<<"\n Working Hours:"<<wh;
cout<<"\n";
}
void calculate(int rate=300)
{
cout<<"\n salary of VisitingStaff is Rs.:- "<<wh*rate;
}
};
int main()
```

```
{

VisitingStaff s[10];
 int i,n;
cout<<"\n How many records you want?\n";
cin>>n;
 for(i=0;i<n;i++)
 {
s[i].getdata();
 }

 for(i=0;i<n;i++)
 {
s[i].display();
}
  for(i=0;i<n;i++)
 {
s[i].calculate();
  }
 return 0;
}
```

1 B) Create a C++ class Employee with data members Emp_Id, Emp_Name, Mobile_No, Salary. Write necessary member functions for the following:
 i.Accept details of n employees
 ii.Display employee details in descending order of their salary.
 iii.Display details of a particular employee.
 (Use Array of object and Use appropriate manipulators)*/

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

 void searchemployee();
 char n[10],c[10];
 long pno;
 class Employee
 {
public:
```

```
int Emp_Id,Mobile_No,Salary;
char name[40];
 void accept()
{
 cout<<"\n Enter  EmpId:-";
cin>>Emp_Id;
 cout<<"\n Enter name of  Emp:-";
cin>>name;
 cout<<"\n Enter  Mobile_No:-";
cin>>Mobile_No;
cout<<"\n Enter the Salary:-";
cin>>Salary;
}
void sort(Employee &r1,Employee &r2)
{
Employee rt;
if(r1.Salary<r2.Salary)
{
rt=r1;
r1=r2;
r2=rt;
}
} void display()
{
cout<<"\nEmpId:-"<<setw(15)<<Emp_Id<<endl;
cout<<"\n Name of  Emp :-"<<setw(15)<<name<<endl;
cout<<"\n Mobile_No :-"<<setw(10)<<Mobile_No<<endl;
cout<<"\n Salary:-"<<setw(15)<<Salary<<endl;
}
 void searchemployee()
{
 if(strcmp(name,c)==0)
 {
cout<<"\n Empame: "<<name<<"\n Salary.: "<<Salary;
//display_data();
 }
}
 };
 int main()
 {

 Employee t[30];
 int num,ch,Salary,i;
 char cont;
 cout<<"\n 1.Accept & display ";
 cout<<"\n 2.Descending";
 cout<<"\n 3.Search by Employee";
 do {
 cout<<"\n Enter your choice: ";
 cin>>ch;
 switch(ch)
 {
 case 1: cout<<"\n How many records you want to enter: ";
 cin>>num;
 for(int i=0;i<num;i++)
 {
 t[i].accept();
 }
 for(i=0;i<num;i++)
 {
 t[i].display();
 }
 break;
 case 2:
 for(i=0;i<num;i++)
{
for(int j=i+1;j<num;j++)
t[i].sort(t[i],t[j]);
t[i].display();
}
 break;
 case 3: cout<<"\n Enter Employee name: ";
 cin>>c;
 for(i=0;i<num;i++)
 {
 t[i].searchemployee();
 }
 break;
 }
 cout<<"\n Do you want to continue: ";
 cin>>cont;
 }
 while(cont=='Y'||cont=='y');
return 0;
 } }
```