Week 1:

1 question:

```cpp
#include<iostream>
using namespace std;
int main()
{
    int a,n,sum=0,r;
    cout<<"enter the number:\n";
    cin>>a;
    n=a;
    while(a>0)
    {
        r=a%10;
        sum=sum+r*r*r;
        a=a/10;
    }
    if(sum==n)
    {
        cout<<"the number is armstrong number"<<endl;
    }
    else
    {
        cout<<"the number is not a armstrong"<<endl;
    }

    return 0;
}
```

Output:

```
enter the number:
153
the number is armstrong number

enter the number:
143
the number is not a armstrong
```

2 question:

```cpp
#include<iostream>
using namespace std;
int main()
```

```cpp
{
    int a,i,mid,f=0;
    cout<<"enter the number:\n";
    cin>>a;
    mid=a/2;
    for(i=2;i<mid;i++)
    {
        if(a%i==0)
        {
            cout<<"number is not a prime number:"<<endl;
            f=f+1;
        }


    }
    if (f==0)
        {
            cout<<"number is prime number"<<endl;
        }

    return 0;
}
```

Output:

enter the number:
9
number is not a prime number:

enter the number:
5
number is prime number

3rd program:

```cpp
#include<iostream>
using namespace std;
int main()
{
    int a=0,b=1,c,i,n;
    cout<<"enter the nth number:\n";
    cin>>n;
    cout<<a<<"->"<<b<<"->";
    for(i=2;i<n;i++)
    {
    c=a+b;
```

```cpp
        a=b;
        b=c;
        cout<<c<<"->";
        }
        return 0;
}
```

Output:

enter the nth number:
9
0->1->1->2->3->5->8->13->21->

Week 2:

1st program:

```cpp
#include<iostream>
using namespace std;
class person
{
    public:
    int age;
    float salary;
    char* name;
    person()
    {
        age=18;
        salary=80000.00;
        name="xyz";
    }
    person(int a,float b,char* c)
    {
        age=a;
        salary=b;
        name=c;
    }
    person(int a,float b)
    {
        age=a;
        salary=b;
        name="ABC";
    }
```

```cpp
    void get()
    {
        cout<<"enter the age of the person:\n";
        cin>>age;
        cout<<"enter the name of the person:\n";;
        cin>>name;
        cout<<"enter the salary of the person:\n";
        cin>>salary;
    }
    void putdata()
    {
        cout<<"name of the person:"<<name<<endl;
        cout<<"age of the person:"<<age<<endl;
        cout<<"salary of the person:"<<salary<<endl;
    }
};
int main()
{
    person p;   //to invoke the default constructor
    person p(19,90000.00,"pavan");  //to invoke parameterized constructor
    person p(19,85000.55);  //to invoke default vth parameterized constructor
    p.get();
    p.putdata();
    return 0;
}
```

Output;
Case 1:
name of the person:xyz
age of the person:18
salary of the person:80000

Case 2:
name of the person:pavan
age of the person:19
salary of the person:90000

Case 3:
name of the person:ABC
age of the person:19
salary of the person:85000.5

2nd program:
```cpp
#include<iostream>
```

```cpp
using namespace std;
inline int add(int a,int b)
{
    return a+b;
}
inline int sub(int a,int b)
{
    return a>b?a-b:b-a;
}
inline int cube(int a)
{
    return a*a*a;
}
inline int mul(int a,int b)
{
    return a*b;
}
int main()
{
    cout<<"addition of the two numbers:"<<add(3,4)<<endl;
    cout<<"subtraction of the two numbers:"<<sub(7,18)<<endl;
    cout<<"cube of the number:"<<cube(9)<<endl;
    cout<<"multiplication of the two numbers:"<<mul(6,7)<<endl;
    return 0;
}
```

Output:

addition of the two numbers:7
subtraction of the two numbers:11
cube of the number:729
multiplication of the two numbers:42

Week 3:

1st program:

```cpp
#include<iostream>
using namespace std;
class test
{
    int a;
```

```cpp
    public:
    friend int value(test t);
    test(int x)
    {
        a=x;
    }
};
int value(test t)
{
    return t.a;
}
int main()
{
    test n(7);

    cout<<"the value is:"<<value(n);
    return 0;

}
```

Output:

the value is:7

2nd program:
```cpp
#include<iostream>
using namespace std;
class test2;
class test1
{

    public:
    int a;
    friend int difference(test1 t1,test2 t2);
    test1(int x)
    {
        a=x;
    }
};
class test2
{
    public:
    int b;
    friend int difference(test1 t1,test2 t2);
    test2(int y)
```

```cpp
    {
        b=y;
    }

};

int difference(test1 t1,test2 t2)
{
    return t1.a>t2.b?t1.a-t2.b:t2.b-t1.a;
}
int main()
{
    test1 t1(78);
    test2 t2(90);
    cout<<"the difference two function:"<<difference(t1,t2);
    return 0;

}
```

Output:

the difference two function:12

3rd question:

```cpp
#include<iostream>
using namespace std;
class student
{
    int roll_no;
    char name[20];
    public:
    void get()
    {
        cout<<"enter the name of the student:\n";
        cin>>name;
        cout<<"enter the roll_no of the student:\n";;
        cin>>roll_no;

    }
    void putdata()
    {
        cout<<"name of the student:"<<name<<endl;
        cout<<"roll_no of the student:"<<roll_no<<endl;

    }
```

```cpp
};
int main()
{
    student p[5];
    cout<<"enter the student info:"<<endl;
    for(int i=0;i<5;i++)
    {
        p[i].get();
    }
    cout<<"student information:"<<endl;
    for(int i=0;i<5;i++)
    {
        p[i].putdata();
    }
    return 0;
}
```

Output:

enter the student info:

enter the name of the student:
pavan
enter the roll_no of the student:
1102
enter the name of the student:
kumar
enter the roll_no of the student:
1103
enter the name of the student:
rakesh
enter the roll_no of the student:
1104
enter the name of the student:
janardhan
enter the roll_no of the student:
1105
enter the name of the student:
vamsi
enter the roll_no of the student:
1106

student information:

name of the student:pavan

roll_no of the student:1102
name of the student:kumar
roll_no of the student:1103
name of the student:rakesh
roll_no of the student:1104
name of the student:janardhan
roll_no of the student:1105
name of the student:vamsi
roll_no of the student:1106

4th question:

```cpp
#include<iostream>
using namespace std;
class book
{   public:
    int bookno;
    char* bookname;
    char* author;
    float price;

    book(int a,char* b,char* c,float d)
    {
        bookno=a;
        bookname=b;
        author=c;
        price=d;
    }

    void display()
    {
        cout<<"no of the book:"<<bookno<<endl;
        cout<<"name of the book:"<<bookname<<endl;
        cout<<"author of the book:"<<author<<endl;
        cout<<"price of the book:"<<price<<endl;

    }
};
int main()
{
    book b(11034,"wings of fire","abdul kalam",575);
    b.display();
    return 0;
}
```

Output:

no of the book:11034
name of the book:wings of fire
author of the book:abdul kalam
price of the book:575

WEEK 4

1 question:

```cpp
#include<iostream>
using namespace std;
#include<stdio.h>
class a
{
    public:
    int x,y;
    void setdata(int a,int b)
    {
        x=a;
        y=b;

    }

};
class b:public a
{
    public:
    int z;
    void assigndata(int c)
    {
        z=c;
    }
    void display()
    {
        cout<<"the values from classA are:"<<x<<"and"<<y<<endl<<"and from classB
is:"<<z;

    }
};
int main()
{
```

```cpp
    b obj;
    obj.setdata(40,50);
    obj.assigndata(90);
    obj.display();
    return 0;
}
```

Output :
the values from classA are:40and50
and from classB is:90

2 question:
```cpp
#include<iostream>
using namespace std;
#include<stdio.h>
class Q
{
    public:
    int q;
    Q()
    {
      q=10;
    }
    void display2()
    {
        cout<<"the value of q :"<<q<<endl;
    }
};
class R:public Q
{
    public:
    int r;
    R()
    {
      r=20;
    }
    void display1()
    {
        cout<<"the value of r:"<<r<<endl;
    }
};
class S:public R
{
    public:
    int s;
    S()
    {
```

```cpp
        s=30;
    }
    void display()
    {
        cout<<"the value of s:"<<s<<endl;
    }
};
int main()
{
    S obj;
    obj.display1();
    obj.display();
    obj.display2();
    return 0;
}
```

Output :
the value of r:20
the value of s:30
the value of q :10

3.question:
```cpp
#include<iostream>
using  namespace std;
#include<stdio.h>
class person
{   public:
    char* fname;
    char* lname;
    void show(char* a,char* b)
    {
        fname=a;
        lname=b;
    }
};
class employe:public person
{
    public:
    int eno;
    float esal;
    char* edept;
    void show2(int a,float b,char* c)
```

```cpp
{
    eno=a;
    esal=b;
    edept=c;
}
void display()
{
    cout<<"first name of the employee="<<fname<<endl;
    cout<<"last of the employee="<<lname<<endl;
    cout<<"the no of the employee is="<<eno<<endl;
    cout<<"the dept ="<<edept<<endl;
    cout<<"the salary of the employe="<<esal<<endl;
}
};
int main()
{
    employe e;
    char fname[30];
    char lname[30];
    char edept[30];
    int eno;
    float esal;
    cout<<"enter your name"<<endl;
    cin>>fname>>lname;
    e.show(fname,lname);
    cout<<"enter your eno &edept & esal"<<endl;
    cin>>eno>>edept>>esal;
    e.show2(eno,esal,edept);
    e.display();
    return 0;
}
```

 Output :

enter your name
pavan
kumar
enter your eno &edept & esal
74
IT
75000
first name of the employee=pavan
last of the employee=kumar
the no of the employee is=74
the dept =IT

the salary of the employe=75000

4.question:

```cpp
#include<iostream>
using namespace std;
#include<stdio.h>
class person
{   public:
    char* fname;
    char* lname;
    void show(char* a,char* b)
    {
        fname=a;
        lname=b;
    }
};
class employe:public person
{
    public:
    int eno;
    float esal;
    char* edept;
    void show2(int a,float b,char* c)
    {
        eno=a;
        esal=b;
        edept=c;
    }

};
class department:public employe
{
    public:
    int dno,experience;
    char* dname;

    void show3(int a,char* b,int c)
    {
        dno=a;
        dname=b;
        experience=c;
```

```cpp
        }
        void display()
        {
            cout<<"first name of the employee"<<fname<<endl;
            cout<<"last name of the employee"<<lname<<endl;
            cout<<"the no of the employee is="<<eno<<endl;
            cout<<"the dept ="<<edept<<endl;
            cout<<"the salary of the employe="<<esal<<endl;
            cout<<"the department number is="<<dno<<endl;
            cout<<"the department name is="<<dname<<endl;
            cout<<"experience is="<<experience<<endl;
        }

};
int main()
{
    department d;
    char fname[30];
    char lname[30];
    char edept[30];
    int eno;
    float esal;
    cout<<"enter your name"<<endl;
    cin>>fname>>lname;
    d.show(fname,lname);
    cout<<"enter your eno &edept & esal"<<endl;
    cin>>eno>>edept>>esal;
    d.show2(eno,esal,edept);
    d.show3(21,"information technology",5);
    d.display();
    return 0;
}
```

OUTPUT:
enter your name
XYZ
ABC
enter your eno &edept & esal
1101
IT
85600
first name of the employeeXYZ
last name of the employeeABC
the no of the employee is=1101
the dept =IT

the salary of the employe=85600
the department number is=21
the department name is=information technology
experience is=5

5.question

```cpp
#include<iostream>
using namespace std;
#include<stdio.h>
class employee
{

    public:
    char* address;
    char* name;
    char* gender;
    int age;
    employee()
    {
     name="pavan";
     address="vijayawada";
     gender="male";
     age=18;
    }
    void display()
    {
       cout<<"name="<<name<<endl;
       cout<<"gender="<<gender<<endl;
       cout<<"addres="<<address<<endl;
       cout<<"age="<<age<<endl;
    }

};
class

 fulltime:public employee
{

    public:
    int salary;
    char* designation;
```

```cpp
    fulltime()
    {
       salary=9867454;
       designation="manager";
    }
    void show()
    {

       cout<<"salary is :"<<salary<<endl;
       cout<<"designation is :"<<designation<<endl;
    }
};
class parttime:public employee
{
    public:
    int workinghrs,rateperhr,amount_pay;
    parttime()
    {
       workinghrs=7;
       rateperhr=5;
    }
    void calculate()
    {
      amount_pay=workinghrs*rateperhr;

    }

    void show2()
    {
       cout<<"amount payable is :"<<amount_pay<<endl;
    }
};
int main()
{
    parttime p;
    fulltime f;
    employee e;
    e.display();
    f.show();
    p.calculate();
    p.show2();

    return 0;
}
```

<u>Output :</u>

name=pavan
gender=male
address=vijayawada
age=18
salary is :9867454
designation is :manager
amount payable is :35

6.question:

```cpp
#include<iostream>
using namespace std;
#include<stdio.h>
class employer
{

    public:
    char* company_name;
    char* city;
    void companydetails(char* a,char* b)
    {
        company_name=a;
        city=b;
    }
    void showdetails()
    {
        cout<<"company_name="<<company_name<<endl;
        cout<<"city="<<city<<endl;
    }
};
class employee:public employer
{
    public:
    int eno;
    float esal;
    char* ename;
    employee(int a,float b,char* c)
```

```cpp
    {
        eno=a;
        esal=b;
        ename=c;
    }
    void show()
    {
        cout<<"the employee number"<<eno<<endl;
        cout<<"the salary of the employee"<<esal<<endl;
        cout<<"the name of the employee"<<ename<<endl;
    }
};
int main()
{
    employee e(74,984354,"pavan kumar");
    e.companydetails("microsoft","hyderabad");

    e.showdetails();
    e.show();
    return 0;
}
```

Output :
company_name=microsoft
city=hyderabad
the employee number74
the salary of the employee984354
the name of the employee pavan kumar

7.question
```cpp
#include<iostream.h>
#include<stdio.h>
class person
{
    char SSN[100];
    char name[100];
    public:
    void getdata()
    {
        cout<<"enter the SSN"<<endl;
        cin>>SSN;
        cout<<"enter the name"<<endl;
        cin>>name;
```

```cpp
    }
    void display()
    {
        cout<<"the SSN is "<<SSN<<endl;
        cout<<"the name is"<<name<<endl;
    }

};
class student:public person
{

    public:
    int roll_no;
    char branch[100];
    float mark1,mark2,mark3;
    void setdata()
    {
        cout<<"enter the roll_no and branch"<<endl;
        cin>>roll_no>>branch;
        cout<<"enter the marks in three subjects"<<endl;
        cin>>mark1>>mark2>>mark3;
    }
    void show()
    {
        cout<<"the roll_no and branch of a student"<<roll_no<<" &"<<branch<<endl;
        cout<<"the marks "<<mark1<<"  &"<<mark2<<" &"<<mark3<<endl;
    }
};
class grade:public student
{
    public:
    float grde;
    void show2()
    {
        grde=(mark1+mark2+mark3)/3;
        if (grde>=90)
        {
            cout<<"A grade"<<endl;
        }124111112
        else if(grde>=80)
        {
            cout<<"B grade"<<endl;
        }
        else if(grde>=70)
        {
            cout<<"C grade"<<endl;
```

```cpp
            }
        else
        {
            cout<<"fail"<<endl;
        }
    }
};
int main()
{
    grade g;
    g.getdata();
    g.setdata();
    g.display();
    g.show();
    g.show2();
    return 0;
}
```

OUTPUT:
enter the SSN=
XYZ
enter the name=
ABCD
enter the roll_no and branch=
110
IT
enter the marks in three subjects=
95
85
80
the SSN is =XYZ
the name is=ABCD
the roll_no and branch of a student110 & IT
the marks 95 &85 &80
B grade

8.question:
```cpp
#include<iostream.h>
#include<stdio.h>
class person
{
```

```cpp
    char fname[100];
    char lname[100];
    public:
    void set()
    {
        cout<<"enter the first name "<<endl;
        cin>>fname;
        cout<<"enter the lname"<<endl;
        cin>>lname;
    }
    void display()
    {
        cout<<"the name of the person="<<this->fname<<endl;
        cout<<"the last name of the person="<<this->lname<<endl;
    }
};
class staff:public person
{

    public:
    int sno;
    int experience;
    void setstaff(int a,int b)
    {
        sno=a;
        experience=b;
    }
    void displaystaff()
    {
        cout<<"sno of the employee"<<sno<<endl;
        cout<<"experience="<<experience<<endl;
    }
};
class employee:public person
{
    public:
    int eno;
    float esal;
    char* edesignation;
    void setemployee(int a,float b,char* c)
    {
        eno=a;
        esal=b;
        edesignation=c;
    }
```

```cpp
void displayemployee()
{
    cout<<"the employee number"<<eno<<endl;
    cout<<"the salary of the employee"<<esal<<endl;
    cout<<"the desgination of the employee"<<edesignation<<endl;
}
};
int main()
{
    employee e;
    e.setemployee(74,836574,"manager");
    e.set();
    staff s;
    s.setstaff(54,5);
    e.display();
    e.displayemployee();
    s.displaystaff();

    return 0;
}
```

Output :

enter the first name
pavan
enter the lname
kumar
the name of the person=pavan
the last name of the person=kumar
the employee number=74
the salary of the employee=836574
the designation of the employee=manager
sno of the employee=54
experience=5

WEEK 5

1st program:

```cpp
#include<iostream>
using namespace std;
class Test
{
    public:
        int a; //only one datamember in the class
        void get()
        {
            cin>>a;       }
        void compare(Test t2) //compare implicit obj and explicit obj
        {
            if(a==t2.a) cout<<"both objs are same";
            else cout<<"both objs are not same";
        }
};
int main()
{
Test t1,t2;
cout<<"enter the value for t1 obj";
t1.get();
cout<<"enter the value for t2 obj";
t2.get();
t1.compare(t2);
return 0;
}
```

Output;
enter the value for t1 obj2
enter the value for t2 obj2
both objs are same

enter the value for t1 obj5
enter the value for t2 obj2
both objs are not same

2rd program:

```cpp
#include<iostream>
using namespace std;
class Test
{
    public:
        int a;
```

```cpp
        void get()
        {
                cin>>a;         }


        void operator ==(Test t2)
        {
                if(a==t2.a) cout<<"both objs are same";
                else cout<<"both objs are not same";
        }
};
int main()
{
Test t1,t2;
cout<<"enter the value for t1 obj";
t1.get();
cout<<"enter the value for t2 obj";
t2.get();

t1==t2;
return 0;
}
```

Output:
enter the value for t1 obj2
enter the value for t2 obj2
both objs are same

3rd question:
AIM:Operator overloading using menu driven program.:
```cpp
#include<iostream>
using namespace std;
class Test
{
    public:
        int a; //only one datamember in the class
        void get()
        {
                cin>>a;         }
        void operator +(Test t2) //compare implicit obj and explicit obj
        {
            cout<<"addition of two operators:"<<a+t2.a<<endl<<endl;
```

```cpp
                //if(a+t2.a) cout<<"both objs are same";
                //else cout<<"both objs are not same";
        }
        void operator -(Test t2)
        {
            cout<<"subtraction of two objects:"<<a-t2.a<<endl<<endl;
        }
        void operator *(Test t2)
        {
            cout<<"multiplication of two objects:"<<a*t2.a<<endl<<endl;
        }
        void operator /(Test t2)
        {
            cout<<"division of two objects:"<<a/t2.a<<endl<<endl;
        }
        void operator %(Test  t2)
        {
            cout<<"modulus division of two objects:"<<a%t2.a<<endl<<endl;
        }
};
int main()
{ int ch;
Test t1,t2;
cout<<"enter the value for t1 obj:";
t1.get();
cout<<"enter the value for t2 obj:";
t2.get();
do{
cout<<"1. addition \n2.subtraction \n3.multiplication \n4.division \n5.modulus division
\n6.EXIT"<<endl;
cin>>ch;
switch(ch)
{
    case 1:
    {
      t1.operator +(t2);
      break;
    }
    case 2:
    {
       t1.operator -(t2);
       break;
    }
    case 3:
    {
       t1.operator *(t2);
```

```cpp
            break;
        }
        case 4:
        {
            t1.operator /(t2);
            break;
        }
        case 5:
        {
            t1.operator %(t2);
            break;
        }
        case 6:
        {
            cout<<"EXIT"<<endl<<endl;;
            break;
        }
        default:
        cout<<"invalid option"<<endl;
    }
}while(ch!=6);

return 0;
}
```

OUTPUT:
enter the value for t1 obj:45
enter the value for t2 obj:25
Case 1
1. addition
2.subtraction
3.multiplication
4.division
5.modulus division
6.EXIT
1
addition of two operators:70
Case 2
1. addition
2.subtraction
3.multiplication
4.division
5.modulus division
6.EXIT
2
subtraction of two objects:20

Case 3

1. addition
2.subtraction
3.multiplication
4.division
5.modulus division
6.EXIT
3
multiplication of two objects:1125
Case 4

1. addition
2.subtraction
3.multiplication
4.division
5.modulus division
6.EXIT
4
division of two objects:1
Case 5

1. addition
2.subtraction
3.multiplication
4.division
5.modulus division
6.EXIT
5
modulus division of two objects:20
Case 6:
1. addition
2.subtraction
3.multiplication
4.division
5.modulus division
6.EXIT
6
EXIT

Week 6:

1st program:

```cpp
#include<iostream>
#include<stack>
using namespace std;
int main()
{
    stack<int> s;
    stack<int> n;
    int x,c;
    cout<<"1.push"<<endl<<"2.pop"<<endl<<"3.display"<<endl<<"4.exit"<<endl;
    while(1)
    {cout<<"Enter the choice"<<endl;
    cin>>c;
    switch(c)
    {
    case 1:
        cout<<"Enter the value to push"<<endl;
        cin>>x;
        s.push(x);
        break;
    case 2:
        s.pop();
        break;
    case 3:
        cout<<"The stack elements are"<<endl;
      n=s;
       while (!n.empty())
       {
        cout << ' ' << n.top();
        n.pop();
       }
        break;
    case 4:
        exit(0);
    default:
        cout<<"invalid";
    }
    }
return 0;
}
```

Output:

1.push
2.pop
3.display
4.exit

Enter the choice
1
Enter the value to push
5
Enter the choice
1
Enter the value to push
6
Enter the choice
1
Enter the value to push
8
Enter the choice
1
Enter the value to push
9
Enter the choice
2
Enter the choice
3
The stack elements are
 8 6 5Enter the choice
4

2nd program;

```cpp
#include<iostream>
#include<stack>
using namespace std;
bool isPalindrome(string s)
{
    int length = s.size();
    stack<char> st;
    int i, mid = length / 2;
    for (i = 0; i < mid; i++)
    {
        st.push(s[i]);
    }
    if (length % 2 != 0) {
        i++;
    }
    char ele;
    while (s[i] != '\0')
    {
        ele = st.top();
        st.pop();
```

```cpp
    if (ele != s[i])
        return false;
        i++;
    }
return true;
}
int main()
{
    string s;
    cout<<"Enter string"<<endl;
    cin>>s;

    if (isPalindrome(s))
    {
        cout << "Yes";
    }
    else {
        cout << "No";
    }

    return 0;
}
```

Output;
case1:
Enter string
civil
No

Case2;
Enter string
civic
Yes

3rd question:
```cpp
#include<iostream>
#include<stack>
using namespace std;
void findMax(stack<int> s)
{
    int m = s.top();

    int a;

    while (!s.empty())
    {
```

```cpp
        a = s.top();

        if (m < a)
            m = a;

        s.pop();
    }

    cout << "\n\nThe maximum element of the Stack is: " << m << endl;
}

int main()
{
    stack<int> s;
    stack<int> n;
    int x,c;

cout<<"Menu"<<endl<<"1.push"<<endl<<"2.pop"<<endl<<"3.Maximum"<<endl<<"4.exit
"<<endl;
    while(1)
    {
        cout<<"Enter the choice"<<endl;
    cin>>c;
    switch(c)
    {
    case 1:
        cout<<"Enter the value to push"<<endl;
        cin>>x;
        s.push(x);
        break;
    case 2:
        s.pop();
        break;
    case 3:

      n=s;
        findMax(s);
        break;
    case 4:
        exit(0);
    default:
        cout<<"invalid";
    }
    }
return 0;
}
```

<u>Output;</u>

```
Menu
1.push
2.pop
3.Maximum
4.exit
Enter the choice
1
Enter the value to push
4
Enter the choice
1
Enter the value to push
5
Enter the choice
1
Enter the value to push
8
Enter the choice
1
Enter the value to push
9
Enter the choice
1
Enter the value to push
4
Enter the choice
1
Enter the value to push
2
Enter the choice
2
Enter the choice
3


The maximum element of the Stack is: 9
Enter the choice
4
```

<u>4th question;</u>

```cpp
#include <iostream>
using namespace std;
class A {
```

```cpp
public:
    int a;
    A() // constructor
    {
        a = 10;
    }
};

class B : public virtual A {
};

class C : public virtual A {
};

class D : public B, public C {
};

int main()
{
    D object; // object creation of class d
    cout << "a = " << object.a << endl;

    return 0;
}
```

Output:

a = 10

Week 7
1st program: (multilevel inheritance)

```cpp
#include<iostream>
using namespace std;
class A
{
    public:
    A()
    {
        cout<<"this is class A constructor."<<endl;
    }
};
class B:public A
{
```

```cpp
   public:
   B()
   {
      cout<<"this is class B constructor."<<endl;
   }
};
class C:public B
{
   public:
   C()
   {
      cout<<"this is class C constructor."<<endl;
   }
};
int main()
{
   C obj;
}
```

Output:
this is class A constructor.
this is class B constructor.
this is class C constructor.

Note:
constructor always starts execution from top to bottom
And Destructors always starts execution from bottom to top

2nd program: (multiple inheritance)
```cpp
#include<iostream>
using namespace std;
class A
{
   public:
   A()
   {
      cout<<"this is class A constructor."<<endl;
   }
};
class B
{
   public:
   B()
   {
      cout<<"this is class B constructor."<<endl;
```

```
    }
};
class C:public A,public B
{
    public:
    C()
    {
        cout<<"this is class C constructor."<<endl;
    }
};
int main()
{
    C obj;

}
```

Output:

this is class A constructor.
this is class B constructor.
this is class C constructor.

3rd question: (hierarchical inheritance)

```
#include<iostream>
using namespace std;
class A
{
    public:
    A()
    {
        cout<<"this is class A constructor."<<endl;
    }
};
class B:public A
{
    public:
    B()
    {
        cout<<"this is class B constructor."<<endl<<endl;
    }
};
class C:public A
{
    public:
    C()
    {
```

```cpp
        cout<<"this is class C constructor."<<endl<<endl;
    }
};
int main()
{
    C obj;
    B ob;
}
```

Output:
this is class A constructor.
this is class C constructor.

this is class A constructor.
this is class B constructor.

4th question:
```cpp
#include<iostream>
using namespace std;
class person
{

    char name[20],gender[20],address[20];
    int age;
    public:
    void get()
    {
        cout<<"enter the name of the person:"<<endl;
        cin>>name;
        cout<<"enter the gender of the person:"<<endl;
        cin>>gender;
        cout<<"enter the address of the person:"<<endl;
        cin>>address;
        cout<<"enter the age of the person:"<<endl;
        cin>>age;
    }
    void display()
    {
        cout<<"the name of the person:"<<name<<endl;
        cout<<"the gender of the person:"<<gender<<endl;
        cout<<"the address of the person:"<<address<<endl;
        cout<<"the age  of the person:"<<age<<endl;
    }
};
class employee:public person
{
```

```cpp
    int deptid;
    char designation[20];
    public:
    void getdata()
    {
        cout<<"enter the id and designation:"<<endl;
        cin>>deptid>>designation;
    }
    void putdata()
    {
        cout<<"the deptid of the person:"<<deptid<<endl;
        cout<<"the designation of the person:"<<designation<<endl;
    }

};
int main()
{
    employee e;
    e.get();
    e.getdata();
    e.display();
    e.putdata();

}
```

Output:

enter the name of the person:
pavan
enter the gender of the person:
male
enter the address of the person:
vijayawada
enter the age of the person:
23
enter the id and designation:
234
manager
the name of the person:pavan
the gender of the person:male
the address of the person:vijayawada
the age of the person:23
the deptid of the person:234
the designation of the person:manager

5th question:

```cpp
#include<iostream>
using namespace std;
#include<stdio.h>
class person
{   public:
    char* fname;
    char* lname;
    void show(char* a,char* b)
    {
        fname=a;
        lname=b;
    }
};
class employe:public person
{
    public:
    int eno;
    float esal;
    char* edept;
    void show2(int a,float b,char* c)
    {
        eno=a;
        esal=b;
        edept=c;
    }

};
class department:public employe
{
    public:
    int dno,experience;
    char* dname;

    void show3(int a,char* b,int c)
    {
        dno=a;
        dname=b;
        experience=c;
    }
    void display()
    {
        cout<<"first name of the employee:"<<fname<<endl;
        cout<<"last name of the employee:"<<lname<<endl;
        cout<<"the no of the employee is="<<eno<<endl;
```

```cpp
        cout<<"the dept ="<<edept<<endl;
        cout<<"the salary of the employe="<<esal<<endl;
        cout<<"the department number is="<<dno<<endl;
        cout<<"the department name is="<<dname<<endl;
        cout<<"experience is="<<experience<<endl;
    }

};
class manager:public person
{   public:
    char* designation;
    float msal;

    void getdata(char* a,float b)
    {
        designation=a;
        msal=b;
    }
    void display2()
    {
        cout<<"designation:"<<designation<<endl;
        cout<<"manager sal:"<<msal<<endl;

    }
};
int main()
{
    department d;
    char fname[30];
    char lname[30];
    char edept[30];
    int eno;
    float esal;
    cout<<"enter your name:"<<endl;
    cin>>fname>>lname;
    d.show(fname,lname);
    cout<<"enter your eno &edept & esal"<<endl;
    cin>>eno>>edept>>esal;
    d.show2(eno,esal,edept);
    d.show3(21,"information technology",5);
    manager m;
    m.getdata("manager",34523);

    d.display();
    m.display2();
    return 0;
```

}
# Output:

enter your name:
pavan
kumar
enter your eno & dept & esal
1104
IT
15000
first name of the employee:pavan
last name of the employee:kumar
the no of the employee is=1104
the dept =IT
the salary of the employe=15000
the department number is=21
the department name is=information technology
experience is=5
designation:manager
manager sal:34523

Week 8:

1st question:
```cpp
#include<iostream.h>
#include<stdio.h>
class person
{
   char SSN[100];
   char name[100];
   public:
   void getdata()
   {
      cout<<"enter the SSN"<<endl;
      cin>>SSN;
      cout<<"enter the name"<<endl;
      cin>>name;
   }
   void display()
   {
      cout<<"the SSN is "<<SSN<<endl;
      cout<<"the name is"<<name<<endl;
   }
```

```cpp
};
class student:public person
{

    public:
    int roll_no;
    char branch[100];
    float mark1,mark2,mark3;
    void setdata()
    {
        cout<<"enter the roll_no and branch"<<endl;
        cin>>roll_no>>branch;
        cout<<"enter the marks in three subjects"<<endl;
        cin>>mark1>>mark2>>mark3;
    }
    void show()
    {
        cout<<"the roll_no and branch of a student"<<roll_no<<" &"<<branch<<endl;
        cout<<"the marks "<<mark1<<"  &"<<mark2<<" &"<<mark3<<endl;
    }
};
class grade:public student
{
    public:
    float grde;
    void show2()
    {
        grde=(mark1+mark2+mark3)/3;
        if (grde>=90)
        {
            cout<<"A grade"<<endl;
        }
        else if(grde>=80)
        {
            cout<<"B grade"<<endl;
        }
        else if(grde>=70)
        {
            cout<<"C grade"<<endl;
        }
        else
        {
            cout<<"fail"<<endl;
        }
    }
```

```cpp
};
int main()
{
    grade g;
    g.getdata();
    g.setdata();
    g.display();
    g.show();
    g.show2();
    return 0;
}
```

OUTPUT:
enter the SSN=
XYZ
enter the name=
ABCD
enter the roll_no and branch=
110
IT
enter the marks in three subjects=
95
85
80
the SSN is =XYZ
the name is=ABCD
the roll_no and branch of a student110 & IT
the marks 95 &85 &80
B grade

3rd program:
```cpp
#include<iostream>
using namespace std;
class student
{
    public:
    int sno;
    char* sname;
    char* branch;
    void read(int a,char* b,char* c)
    {
```

```cpp
            sno=a;
            sname=b;
            branch=c;
        }
        void display()
        {
            cout<<"sno:"<<sno<<endl;
            cout<<"sname:"<<sname<<endl;
            cout<<"branch:"<<branch<<endl;
        }

};
class test:virtual public student
{
    public:
    int sem1,sem2,sem3,test_cre;
    void get(int a,int b,int c)
    {
        sem1=a;
        sem2=b;
        sem3=c;
        test_cre=sem1+sem2+sem3;
    }
};
class sports:virtual public student
{
    public:
    int ncc,nss,others,sport_cre;
    void getdata(int a,int b,int c)
    {
        ncc=a;
        nss=b;
        others=c;
        sport_cre=ncc+nss+others;
    }
};
class result:public test,public sports
{
    public:
    int res;
    void show3()
    {
        res=test_cre+sport_cre;
        cout<<"total no_of credits gained by the student :"<<res;
    }
};
```

```
int main()
{
  result r;
  r.read(11,"xyz","it");
  r.display();

  r.get(3,3,2);
  r.getdata(2,2,1);
  r.show3();
  return 0;

}
```

Output:
sno:11
sname:xyz
branch:it
total no_of credits gained by the student :13

Week 9:(Abstract class)
1st question:
a) program:
```
#include<iostream>
using namespace std;
class shape
{
   public:
   float d1,d2;

   void getdata()
   {   cout<<"enter the dimensions:"<<endl;
      cin>>d1>>d2;
   }
};
class triangle:public shape
{
   public:
```

```cpp
        void display()
        {
            cout<<"area of the triangle:"<<0.5*d1*d2<<endl;
        }
};
class rectangle:public shape
{
    public:
    void display()
    {
        cout<<"the area of rectangle:"<<d1*d2<<endl;
    }
};
int main()
{
    triangle t;
    t.getdata();
    t.display();
    rectangle r;
    r.getdata();
    r.display();
    return 0;
}
```

Output:

enter the dimensions:
25 10
area of the triangle:125
enter the dimensions:
22 15
the area of rectangle:330

b) program:

```cpp
#include<iostream>
using namespace std;
class shape
{
    public:
    float d1,d2;
    public:
    void getdata()
    {   cout<<"enter the dimensions:"<<endl;
```

```cpp
        cin>>d1>>d2;
    }
virtual float area()=0;
};
class triangle:public shape
{
    public:
    virtual float area()
    {
        cout<<"area of the triangle:"<<0.5*d1*d2<<endl;
    }
};
class rectangle:public shape
{
    public:
    virtual float area()
    {
        cout<<"the area of rectangle:"<<d1*d2<<endl;
    }
};
int main()
{
    triangle t;
    t.getdata();
    t.area();
    rectangle r;
    r.getdata();
    r.area();
    return 0;
}
```

Output;

enter the dimensions:
23 45
area of the triangle:517.5
enter the dimensions:
22 44
the area of rectangle:968

2nd question:

```cpp
#include<iostream>
using namespace std;
class employee
{
    public:
    int month,wk,pcs,hrs;
    virtual float payroll()=0;
};
class boss:public employee
{
    public:
    virtual float payroll()
    {   cout<<"enter the no_of months"<<endl;
        cin>>month;
        cout<<"the payment of the boss:"<<50000*month<<endl; //50k per month
    }

};
class week:public employee
{   public:
    virtual float payroll()
    {   cout<<"enter the no_of weeks"<<endl;
        cin>>wk;
        cout<<"the payment of the weekly based workers:"<<10000*wk<<endl; //10k per week
    }

};
class pieces:public employee
{   public:
    virtual float payroll()
    {   cout<<"enter the no_of pieces"<<endl;
        cin>>pcs;
        cout<<"the payment of the pieces wish workers:"<<1000*pcs<<endl; //1k per pieces
    }
};
class hour:public employee
{   public:
    virtual float payroll()
    {   cout<<"enter the no_of hrs"<<endl;
        cin>>hrs;
        cout<<"the payment of the hourly based workers:"<<500*hrs<<endl; //500k per hr
    }
```

```cpp
};
int main()
{
    boss b;
    week w;
    pieces p;
    hour h;
    b.payroll();
    w.payroll();
    p.payroll();
    h.payroll();
    return 0;
}
```

Output:

enter the no_of months
10
the payment of the boss:500000
enter the no_of weeks
7
the payment of the weekly based workers:70000
enter the no_of pieces
5
the payment of the pieces wish workers:5000
enter the no_of hrs
10
the payment of the hourly based workers:5000

3rd question:

```cpp
#include<iostream>
using namespace std;
class convert
{
    public:
    float fh,mtr;
    int hr;
virtual float conversion()=0;
};
class fh_to_cls:public convert
{
    public:
```

```cpp
    virtual float conversion()
    {
    cout<<"enter the fahrenheit value:"<<endl;
    cin>>fh;
    cout<<"fahrenheit to celsius value:"<<(fh-32)*0.5556<<endl;
    }
};
class mtr_to_km:public convert
{
    public:
    virtual float conversion()
    {
    cout<<"enter the meters value:"<<endl;
    cin>>mtr;
    cout<<"meters to kilometer value:"<<mtr/1000<<endl;
    }

};
class hrs_to_sec:public convert
{
    public:
    virtual float conversion()
    {
    cout<<"enter the hrs value:"<<endl;
    cin>>hr;
    cout<<"hours to second value:"<<hr*3600<<endl;
    }

};
int main()
{
    fh_to_cls f;
    mtr_to_km m;
    hrs_to_sec h;
    f.conversion();
    m.conversion();
    h.conversion();
    return 0;
}
```

Output;

enter the fahrenheit value:
68
fahrenheit to celsius value:20.0016

enter the meters value:
5000
meters to kilometer value:5
enter the hrs value:
4
hours to second value:14400

4th program:

```cpp
#include<iostream>
using namespace std;
#include<stdio.h>
class employee
{

    public:
    char* address;
    char* name;
    char* gender;
    int age;
    employee()
    {
     name="pavan";
     address="vijayawada";
     gender="male";
     age=18;
    }
    void display()
    {
       cout<<"name="<<name<<endl;
       cout<<"gender="<<gender<<endl;
       cout<<"addres="<<address<<endl;
       cout<<"age="<<age<<endl;
    }

};
class

 fulltime:public employee
{

    public:
    int salary;
    char* designation;
    fulltime()
```

```cpp
    {
        salary=9867454;
        designation="manager";
    }
    void show()
    {

        cout<<"salary is :"<<salary<<endl;
        cout<<"designation is :"<<designation<<endl;
    }
};
class parttime:public employee
{
    public:
    int workinghrs,rateperhr,amount_pay;
    parttime()
    {
        workinghrs=7;
        rateperhr=5;
    }
    void calculate()
    {
      amount_pay=workinghrs*rateperhr;

    }

    void show2()
    {
        cout<<"amount payable is :"<<amount_pay<<endl;
    }
};
int main()
{
    parttime p;
    fulltime f;
    employee e;
    e.display();
    f.show();
    p.calculate();
    p.show2();

    return 0;
}
```

Output :

name=pavan
gender=male
address=vijayawada
age=18
salary is :9867454
designation is :manager
amount payable is :35

Week 10:

1st program:

```cpp
#include<iostream>
using namespace std;
template <typename t1,typename t2,typename t3>
t1 add(t1 a,t2 b,t3 c)
{
    return a+b+c;
}
int main()
{
    cout<<"the addition of numbers:"<<add(4.5,7,0.54)<<endl;    //adding 3 types of data types
    cout<<"the addition of numbers:"<<add(7,3,4)<<endl;          //adding integer values
    cout<<"the addition of numbers:"<<add(1.5,2.5,3.5)<<endl;    //adding float values
    return 0;
}
```

Output:
the addition of numbers:12.04
the addition of numbers:14
the addition of numbers:7.5

2nd question:

```cpp
#include<iostream>
using namespace std;
template <typename t1,typename t2>
```

```cpp
void change(t1 a,t2 b)
{
  a=a+b;
  b=a-b;
  a=a-b;
  cout<<"the elements after swapping:"<<a<<" "<<b<<endl;
}
int main()
{
    change(4.5,6.5);  //swapping of float values
    //change(4,6.5);    //dissimilar data types are not allowed  for swapping
    change(4,6);    //swapping of integer values
    return 0;
}
```

Output:

the elements after swapping:6.5 4.5
//the elements after swapping:6 3.5 //this is not possible
the elements after swapping:6 4

3rd question:

```cpp
#include <iostream>
using namespace std;
template <typename T>
void bubbleSort(T a[], int n) {
        for (int i = 0; i < n - 1; i++)
                for (int j = n - 1; i < j; j--)
                        if (a[j] < a[j - 1])
                        swap(a[j], a[j - 1]);
}
int main() {
        int a[5] = {10, 50, 30, 40, 20};
        int n = sizeof(a) / sizeof(a[0]);


        bubbleSort(a, n);

        cout << "integer Sorted array : ";
        for (int i = 0; i < n; i++)
                cout << a[i] << " ";
        cout << endl;
```

```cpp
    float b[4] = {4.5, 6.5, 1, 0};
    int n1 = sizeof(b) / sizeof(b[0]);
bubbleSort(b, n1);
    cout << "float Sorted array : ";
    for (int i = 0; i < n1; i++)
            cout << b[i] << " ";
    cout << endl;
return 0;
}
```

Output:
integer Sorted array : 10 20 30 40 50
float Sorted array : 0 1 4.5 6.5

4th question:
```cpp
#include<iostream>
using namespace std;
template <typename t1,typename t2>
t1 change(t1 a,t2 b)
{
    return a>b?a:b;
}
int main()
{
   cout<<"the maximum number among two"<<change(4.5,7)<<endl; //greater
among int and float
   cout<<"the maximum number among two"<<change(4.5,7.8)<<endl;  //greater
among two float values
   cout<<"the maximum number among two"<<change(9,3)<<endl;  //greater
among two int values
   return 0;
}
```

Output:
the maximum number among two 7
the maximum number among two 7.8

the maximum number among two 9

5th question:

Here we overloaded add and mul functions
 Same name but different type of args:
Program:
a) addition

```
#include<iostream>
using namespace std;
template <typename t1>
t1 add(t1 a,t1 b)
{
        cout<<"\ninvoked if the argument is other than double...\n";

    return a+b;
}
double add(double c,double d)
{
    cout<<"\ninvoked if the argument is double...\n";
    return c+d;
}
int main()
{   cout<<"addition of the number:"<<add(4,8)<<endl;   //int data type
    cout<<"addition of the number:"<<add(7.5,8.7)<<endl;   //float data type
    return 0;
}
```

Output:
addition of the number:
invoked if the argument is other than double...
12
addition of the number:
invoked if the argument is double...
16.2

b) multiplication:

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

```cpp
template <typename t1>
t1 mul(t1 a,t1 b)
{
    cout<<"\ninvoked if the argument is other than double...\n";

    return a*b;
}
double mul(double c,double d)
{
    cout<<"\ninvoked if the argument is double...\n";
    return c*d;
}
int main()
{   cout<<"multiplication of the number:"<<mul(4,8)<<endl;   //int data type
    cout<<"multiplication of the number:"<<mul(7.5,8.7)<<endl;   //float data type
    return 0;
}
```

Output:

multiplication of the number:
invoked if the argument is other than double...
32
multiplication of the number:
invoked if the argument is double...
65.25

TEMPLATES CLASS FUNCTION;

Generic Templates class Templates

1   Write a c++ program using class template for linear search

```cpp
#include<iostream>
#include<array>
using namespace std;
template <class t1,class t2>
class test
{
    t1* a;
    t2 b;
```

```cpp
    t2 n;
    public:
    test(t1* x,t2 y,t2 z)
    {
        a=x;
        b=y;
        n=z;

    }
    void linear()
    {

        int d=0;

        for(int i=0;i<n;i++)
        {
            if (a[i]==b)
            {
                d+=1;
                cout<<"the element is found at:"<<i<<"th position"<<endl;
            }

        }
        if(d==0)
        {
            cout<<"element not found"<<endl;
        }

    }
};
int main()
{   int a[5]={1,2,4,5,6};
    int n=sizeof(a)/sizeof(a[0]);
    test <int,int>t (a,6,n);
    t.linear();
    return 0;
}
```

OUTPUT:

the element is found at:4th position

W2 Write a c++ program using class template for Binary search

```cpp
#include<iostream>
#include<array>
using namespace std;
template <class t1,class t2>
class test
{
    t1* a;
    t2 b;
    t2 n;
    public:
    test(t1* x,t2 y,t2 c)
    {
        a=x;
        b=y;
        n=c;
    }
    void binary()
    {

        int c=0,mid,low=0,high=n-1;
        while(low<=high)
        {
            mid=(low+high)/2;
            if(b<a[mid])
            {
                high=mid-1;
            }
            else if(b>a[mid])
            {
                low=mid+1;
            }
            else if(b==a[mid])
```

```cpp
                    {
                        c+=1;
                        cout<<"the element is found at:"<<mid<<"th position"<<endl;
                        break;

                    }
                }

                if(c==0)
                {
                    cout<<"element not found"<<endl;
                }

            }
};
int main()
{   int a[6]={1,2,3,4,5,6};
    int n=sizeof(a)/sizeof(a[0]);
    test <int,int>t (a,4,n);


    t.binary();
    return 0;
}
```

Output:
the element is found at:3th position

3   Write a c++ program using a class template for performing calculator operations.

```cpp
#include<iostream>
using namespace std;
template <class t1,class t2>
class calculator
```

```cpp
{
    t1 a;
    t2 b;
    public:
    calculator(t1 x,t2 y)
    {
        a=x;
        b=y;
    }
    void display()
    {
        cout<<"addition of the two elements:"<<a+b<<endl;
        cout<<"subtraction  of the two elements:"<<a-b<<endl;
        cout<<"multiplication of the two elements:"<<a*b<<endl;
        cout<<"division of the two elements:"<<a/b<<endl;

    }
};
int main()
{
    calculator<int,int> c(50,10);
    c.display();
    return 0;
}
```

Output:

addition of the two elements:60
subtraction of the two elements:40
multiplication of the two elements:500
division of the two elements:5

Week 11

1st program:

```cpp
#include<iostream>
#include<vector>
#include<algorithm>
using namespace std;
int main()
{
    vector<int> a;
    a.push_back(50);
    a.push_back(7);
    a.push_back(45);
    a.push_back(11);
    a.push_back(35);
    a.push_back(25);
    cout<<"elements the vector:\n";
    for (auto i=a.begin();i!=a.end();++i)
    {
        cout<<*i<<" ";
    }
    sort(a.begin(),a.end());
    cout<<"\nelements after sorting:\n";
    for (auto i=a.begin();i!=a.end();++i)
    {
        cout<<*i<<" ";
    }

    return 0;
}
```

Output:

elements the vector:
50 7 45 11 35 25
elements after sorting:
7 11 25 35 45 50

2nd program:

```cpp
#include<iostream>
#include<deque>
#include<algorithm>
using namespace std;
int main()
{
    deque<int> a;
    a.push_front(50);
    a.push_front(7);
    a.push_front(45);
    a.push_front(11);
    a.push_front(35);
    a.push_front(25);
    cout<<"elements the vector:\n";
    for (auto i=a.begin();i!=a.end();++i)
    {
        cout<<*i<<" ";
    }
    sort(a.rbegin(),a.rend());   //this is will sort the elements in descending order
    cout<<"\nelements after sorting:\n";
     for (auto i=a.begin();i!=a.end();++i)
    {
        cout<<*i<<" ";
    }

    return 0;
}
```

Output:

```
elements the vector:
25 35 11 45 7 50
elements after sorting:
50 45 35 25 11 7
```

3rd program:

```cpp
#include<iostream>
#include<deque>
#include<algorithm>
using namespace std;
int main()
{   int key,found=0;
    deque<int> a;
    a.push_front(50);
    a.push_front(7);
    a.push_front(45);
    a.push_front(11);
    a.push_front(35);
    a.push_front(25);
    cout<<"elements the vector:\n";
    for (auto i=a.begin();i!=a.end();++i)
    {
        cout<<*i<<" ";
    }
    cout<<"\nenter the key value:\n";
    cin>>key;
     for (auto i=a.begin();i!=a.end();++i)
    {
        if(*i==key)
        {
            found++;
            cout<<"\nelement "<<*i<<" is found in the vector";
        }
    }
    if(found==0)
    {
        cout<<"\nelement not found";
    }

    return 0;
}
```

Output:
Case 1
elements the vector:

25 35 11 45 7 50

enter the key value:

11

element 11 is found in the vector

Case 2:
elements the vector:

25 35 11 45 7 50

enter the key value:

78

element not found

Week 12

2b explain the process of handling derived class exceptions?

Program:

```
#include<iostream>
using namespace std;
class Base {};
class Derived: public Base {};
int main()
{
  Derived d;
  // some other stuff
  try {
    // Some monitored code
    throw d;
  }
  catch(Base b) {
```

```
        cout<<"Caught Base Exception";
    }
    catch(Derived d) {  //This catch block is NEVER executed
        cout<<"Caught Derived Exception";
    }
    getchar();
    return 0;
}
```

Output:
Caught Base Exception

2b design a solution for handling divide by zero error
Program:

```
#include<iostream>

using namespace std;

int main()
{
int a=10, b=0, c;
try
{
    //if a is divided by b(which has a value 0);
    if(b==0)
            throw(c);
    else
    c=a/b;

}
catch(char c)    //catch block to handle/catch exception
{
    cout<<"Caught exception : char type ";
}
catch(int i)    //catch block to handle/catch exception
{
```

```
        cout<<"Caught exception : int type ";
}
catch(short s)     //catch block to handle/catch exception
{
        cout<<"Caught exception : short type ";
}
cout<<"\n Hello";
}
```

Output:
Caught exception : int type
 Hello

4b Design a catch block to handle multiple exceptions (divide by zero,array out of bounds exception)

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

int main()
{
   int arr[] = {1,2,0,4,5};
   int len,a=4,c,d;
   len = sizeof(arr)/sizeof(arr[0]);
   try{
      for(int i=0;i<=len;i++){
         if(i>len-1){
            throw len;
         }
         // cout<<arr[i]<<" ";
         if(arr[i]==0){
            cout<<"infinity"<<endl;
            d = arr[i];
            throw d;
```

```
        }
        else{
          c = a/arr[i];
          cout<<"div : "<<c<<endl;
        }

      }
    }
    catch(int len){
       cout<<"Array out of bounds exception";
    }
    catch(int d){
       cout<<"Division by zero";
    }

    return 0;
}
```

Output:

div : 4
div : 2
infinity
Array out of bounds exception

6b Design a solution to check if the given phone number is valid or not , if the number has exactly 5 digits exactly display as valid using exceptions handling
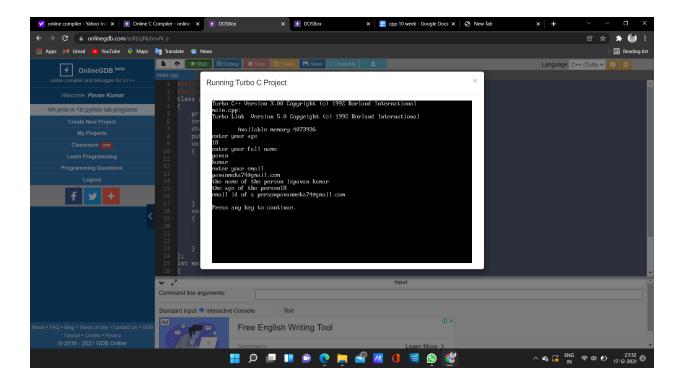
Program:

```
#include<iostream>
#include<string>
using namespace std;
int main()
{  int a,len=0,r,n;
   cout<<"enter the number:";
   cin>>a;
   n=a;
```

```cpp
    while(a>0)
    {
        r=a%10;
        len++;
        a=a/10;

    }
    try
    {
        if(len==5)
        {
            cout<<n<<" is valid number.\n";
        }
        else
        {
            throw len;
        }
    }
    catch(int len)
    {
        cout<<n<<" is not a valid number.\n";
    }
}
```

Output:

enter the number:53265
53265 is valid number.

1st year week 10&11 programs:

1st :
```cpp
#include<iostream>
using namespace std;
class person
{
    public:
    int age;
    string name;
    person(int a,string b)
    {
        age=a;
        name=b;
    }
    void display()
    {
        cout<<"the name and age of the person:"<<age<<" & "<<name<<endl;
    }
};
int main()
{
    person p(18,"pavan");
    p.display();
    return 0;
}
```

2nd:Program:
```cpp
#include<iostream>
using namespace std;
class person
{
    int age;
    string name;
    public:
    void get()
    {
        cout<<"enter your age and name"<<endl;
        cin>>age>>name;
    }
    void display()
    {
        cout<<"the name of the person::"<<name;
        cout<<"\nthe age of the person:"<<age;
    }
};
int main()
{
```

```cpp
        person p;
        p.get();
        p.display();
        return 0;

}
```

3rd program :
```cpp
#include<iostream.h>
#include<stdio.h>
class person
{
    private:
    int age;
    char fname[10],lname[10],email[30];
    public:
    void get()
    {
        cout<<"enter your age"<<endl;
        cin>>age;
        cout<<"enter your full name"<<endl;
        cin>>fname>>lname;
        cout<<"enter your email"<<endl;
        cin>>email;
    }
    void display()
    {
        cout<<"the name of the person is"<<fname<<" "<<lname<<endl;
        cout<<"the age of the person"<<age<<endl;
        cout<<"email id of a person"<<email<<endl;
    }
};
int main()
{
    person p;
    p.get();
    p.display();
    return 0;
}
```

Output:

4th program:
```cpp
#include<iostream.h>
#include<stdio.h>
class abstall
{
   protected:
   float cost;
   char name[20],details[20],owner_name[20];
   public:
   void get1()
   {
      cout<<"enter the details in the order as mentioned"<<endl;
      cout<<"1.name 2.details 3.cost 4.owner_name"<<endl;
      cin>>name;
      cin>>details;
      cin>>cost;
      cin>>owner_name;
   }
   void show1()
   {
      cout<<"stall name: "<<name<<endl;
      cout<<"details: "<<details<<endl;
      cout<<"cost: "<<cost<<endl;
      cout<<"owner_name: "<<owner_name<<endl;
   }
};
```

```cpp
class exstall:public abstall
{
    int no_of_tv_set;
    public:
    void get2()
    {
        cout<<"enter no_of_tv_set:"<<endl;
        cin>>no_of_tv_set;
    }
    void show2()
    {
        cout<<"stall name:"<<name<<endl;
        cout<<"details: "<<details<<endl;
        cout<<"cost: "<<cost<<endl;
        cout<<"owner_name: "<<owner_name<<endl;
        cout<<"no_of_tv_set: "<<no_of_tv_set<<endl;
    }
};
class prestall:public abstall
{
    int no_of_pro;
    public:
    void get3()
    {
        cout<<"enter the no_of projectors: "<<endl;
        cin>>no_of_pro;
    }
    void show3()
    {
        cout<<"stall name: "<<name<<endl;
        cout<<"details: "<<details<<endl;
        cout<<"cost:"<<cost<<endl;
        cout<<"owner_name: "<<owner_name<<endl;
        cout<<"no_of projectors: "<<no_of_pro<<endl;
    }
};
int main()
{
    int ch;

    do{
    cout<<"stall type"<<endl;
    cout<<"1.stall\n2.executive stall\n3.premium stall\n4.exit\n";
    cin>>ch;
    switch(ch)
    {
```

```cpp
            case 1:
            {
                abstall s;
                s.get1();
                s.show1();
                break;
            }
            case 2:
            {
                 exstall e;
                 e.get1();
                 e.get2();
                 e.show2();
                 break;
            }
            case 3:
            {
                prestall p;
                p.get1();
                p.get3();
                p.show3();
                break;
            }
            case 4:
            {
                cout<<"exit"<<endl;
                break;
            }
            default:
                {
                cout<<"invalid operation"<<endl;
                }
        }

    }while(ch!=4);
    return 0;
}
```

Output :

Week 6

AIM:-VIRTUAL BASE CLASS PROGRAM:

CODE:

```cpp
#include<iostream>
using namespace std;
class A
{
    public:
    int a;
};
class B:virtual public A{
    public:
    int b;
};
class C:virtual public A{
    public:
    int c;
};
class D: public B,public C{
    public:
    int d;
};
int main()
{
    D obj;
    obj.a=25;          //statement 1
    obj.a=100;         //statement 2

    obj.b=50;
    obj.c=75;
```

```
    obj.d=150;

    cout<< "\n A : "<< obj.a;
    cout<< "\n B : "<< obj.b;
    cout<< "\n C : "<< obj.c;
    cout<< "\n D : "<< obj.d;
}
```

OUTPUT:
 A : 100
 B : 50
 C : 75
 D : 150

EXPLANATION:
According to the above example, ClassD has only **one copy** of
ClassA, therefore, **statement 2** will **overwrite** the value of a, given at
statement 1.

New and delete operators overloading

Template programs :

```
//design a template function which returns the maximum value among two values
#include<iostream>
using namespace std;
template <typename T>
T great(T a,T b)
{
    return a>b?a:b;
}
int main()
```

```cpp
{
    cout<<great(2,5)<<endl;
    cout<<great(1.4,8.9)<<endl;

    return 0;
}
```

1st program:

```cpp
#include<iostream>
using namespace std;
template <typename t1,typename t2,typename t3>
t1 add(t1 a,t2 b,t3 c)
{
    return a+b+c;
}
int main()
{
    cout<<add(4.5,7,0.54)<<endl;
    return 0;
}
```

Output:
the sum of the all values:12.04

2nd question:

```cpp
#include<iostream>
using namespace std;
template <typename t1,typename t2>
void change(t1 a,t2 b)
{
  a=a+b;
  b=a-b;
  a=a-b;
  cout<<"the elements after swapping:"<<a<<" "<<b<<endl;
}
int main()
{
    change(4.5,6.5);  //addition of float values
    change(4,6.5);    //addition of both integer and float values
    change(4,6);    //addition of integer values
    return 0;
}
```

Output:

the elements after swapping:6.5 4.5
the elements after swapping:6 3.5
the elements after swapping:6 4

3rd question:

```cpp
#include<iostream>
using namespace std;
template <typename t1,typename t2>
t1 change(t1 a,t2 b)
{
    return a>b?a:b;
}
int main()
{
    cout<<"the maximum number among two"<<change(4.5,7)<<endl;
    return 0;
}
```

Output:
the maximum number among two7


```cpp
#include<iostream>
using namespace std;
class bank
{
    public:
    int tym;
    bank(int a)
    {
        tym=a;
    }
    void display()
    {
        if(tym<=2)
        {
            cout<<"interest rate is 5%\n";
```

```cpp
        }
        else if(tym>2&&tym<=4)
        {
            cout<<"interest rate is 8%\n";
        }
        else if(tym>4)
        {
            cout<<"interest rate is 10%\n";
        }
    }
};
int main()
{   int a;
    cout<<"enter the time period:\n";
    cin>>a;
    bank b(a);
    b.display();
    return 0;
}
```