**Dr. B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY JALANDHAR**



**Practical File**

**For**

**DESIGN AND ANALYSIS OF ALGORITHMS**

**(CSX-232)**

**Session - (Jan - May, 2019)**

**Submitted to: Submitted by:**

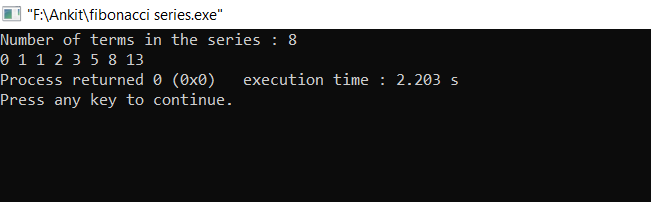
**Dr. Kuldeep Kumar Ankit Goyal**

**Assistant Professor, 17103011**

**Computer Science and Engineering G-1**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Program** | **Date** | **Remarks** |
| **1.** | **Write a program to output the fibonacci series.** |  |  |
| **2.** | **Print a specific pattern.** |  |  |
| **3.** | **Write a program to find smallest element in the array.** |  |  |
| **4.** | **Write a program to print the sum of two integer values with the proper messages** |  |  |
| **5.** | **Write a function to print the factorial of a number.** |  |  |
| **6.** | **Write a program to implement function overloading.** |  |  |
| **7.** | **Write a program to make use of inline function.** |  |  |
| **8.** | **Write a program to implement default parameters where you declare a function with default parameters and you call the function**  **(i). With no default arguments**  **(ii). With 2nd parameter as defaulted**  **(iii). With both parameters as defaulted** |  |  |
| **9.** | **Write a program to add two matrices.** |  |  |
| **10.** | **Write a program to implement swapping function using call by value.** |  |  |
| **11.** | **Write a program to implement swapping function using call by reference.** |  |  |
| **12.** | **Write a program to print the factorial of a number using recursion.** |  |  |
| **13.** | **Create a simple class containing two data items in private and two member functions in public. Make an object of this class and call these two functions.** |  |  |
| **14.** | **Write a program to read and add two times using class.** |  |  |
| **15.** | **Write a program to implement default parameters where you declare a function with default parameters and you call the function**  **(i). With no default arguments**  **(ii). With 2nd parameter as defaulted**  **(iii). With both parameters as defaulted** |  |  |
| **16.** | **Create class set to store the integer values and implement following member functions**  **(i). Is\_member**  **(ii). Union**  **(iii). intersection**  **(iv). minus** |  |  |
| **17.** | **Create a class book to store details of book like book name, author name, and copies count. Create 20 different books.** |  |  |
| **18.** | **Make a class account that represents your bank account. This class should allow you to deposit and withdraw money and show the account details.** |  |  |
| **19.** | **Create a class student to get the details of the students and print them.** |  |  |
| **20.** | **Write a program to create a library that can lend the book and add important details.** |  |  |
| **21.** | **Write a program to take input for sum of complex numbers and display using default constructor.** |  |  |
| **22.** | **Write a program to class string to perform basic functions using constructors.** |  |  |
| **23.** | **Write a program to demonstrate concept of destructor.** |  |  |
| **24.** | **Write a program to demonstrate single inheritance** |  |  |
| **25.** | **Write a program to demonstrate multiple inheritance.** |  |  |
| **26.** | **Write a program to demonstrate multilevel inheritance.** |  |  |
| **27.** | **Write a program to demonstrate hybrid inheritance.** |  |  |
| **28.** | **Write a program to make classes staff, teacher, typist and professor of an educational institute and operate on their databases especially qualifications.** |  |  |
| **29.** | **Write a program to show overloading.**  **(i). Unary**  **(ii). Binary** |  |  |
| **30.** | **Write a program to create a class shape to calculate area of rectangle, triangle and hexagon using inheritance.** |  |  |
| **31.** | **Write a program to create network of the classes (hybrid inheritance) by classes person, account, admin and master.** |  |  |
| **32.** | **Write a program to show the concept of run-time polymorphism using virtual function.** |  |  |
| **33.** | **Write a program to find area of triangle by three different points using containership.** |  |  |
| **34.** | **Improvise the above SHAPE program to display the area of circle.** |  |  |



**Q1. Write a program to output the fibonacci series.**

Ans:

#include “iostream”

using namespace std;

int main()

{

int c,n,p=0,q=1;

cout<<"Number of terms in the series : ";

cin>>n;

if(n>0)

{

if(n==1)

cout<<p;

if(n>=2)

{

cout<<p<<" "<<q<<" ";

n-=2;

while(n)

{

c=p+q;

p=q;

q=c;

cout<<c;

cout<<” “;

n--;

}

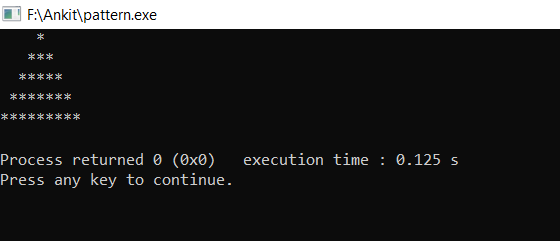
}

}

else

cout<<"not allowed\n"<<endl;

}



**Q2.** **Print a specific pattern.**

#include "iostream"

using namespace std;

int main()

{

int i,j;

for(i=0;i<5;i++)

{

for(j=0;j<4-i;j++)

cout<<" ";

for(j=0;j<2\*i+1;j++)

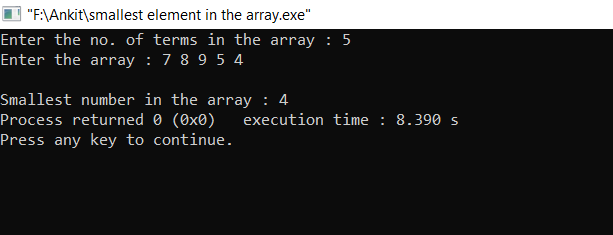
cout<<"\*";

cout<<endl;

}

return 0;

}



**Q3. Write a program to find smallest element in the array.**

Ans:

#include "iostream"

using namespace std;

int main()

{

int p,n,i=0;

cout<<"Enter the no. of terms in the array : ";

cin>>n;

cout<<”Enter the array : “;

cin>>a;

p=a;

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

{

cin>>a;

if(p<a)

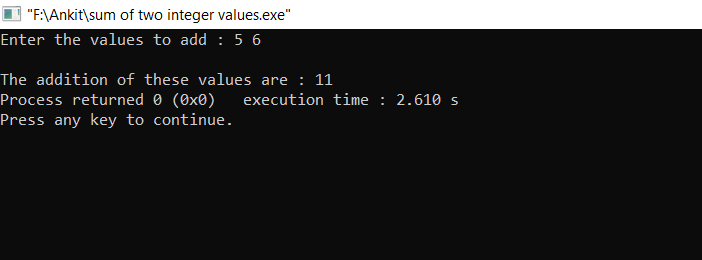
p=a;

}

cout<<"\nSmallest number in the array : "<<p;

return 0;

}



Q4. **Write a program to print the sum of two integer values with the proper messages.**

#include "iostream"

using namespace std;

int main()

{

int p,q;

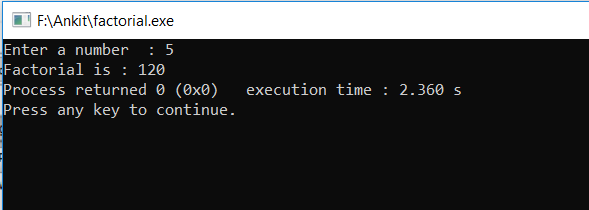
cout<<"Enter the values to add : ";

cin>>p>>q;

cout<<"\nThe addition of these values are : "<<p+q;

return 0;

}



**Q. Write a program to find the factorial of a number.**

#include "iostream"

using namespace std;

int main()

{

int n,i;

cout<<"Enter the value : ";

cin>>n;

i=n;

int ans=1;

while(i)

{

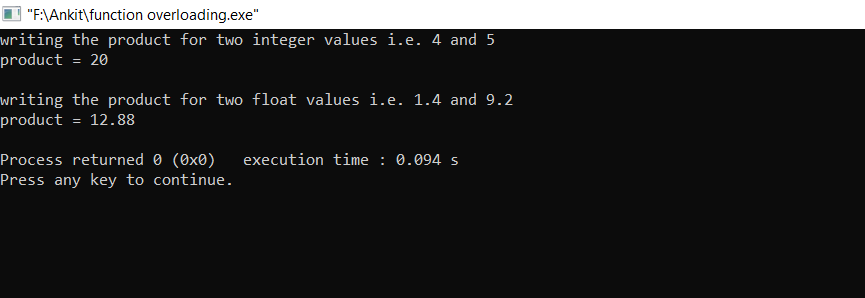
ans\*=i;

i--;

}

cout<<"Factorial to the number "<<n<<" is "<<ans;

}



**Q6. Write a program to implement function overloading.**

#include "iostream"

using namespace std;

void product(int a,int b)

{

cout<<a\*b<<"\n";

}

void product(double a,double b)

{

cout<<a\*b<<"\n";

}

int main()

{

cout<<"writing the product for two integer values i.e. 4 and 5 ";

cout<<"\nproduct = ";

product( 4 , 5 );

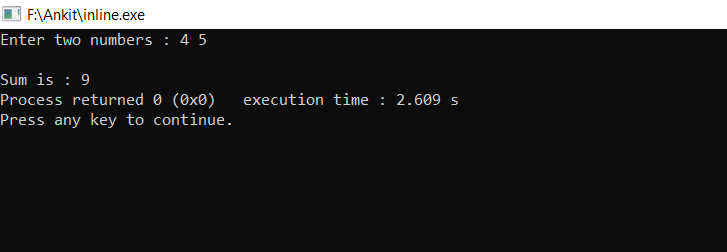
cout<<"\nwriting the product for two float values i.e. 1.4 and 9.2”;

cout<<"\nproduct = ";

product( 1.4 , 9.2 );

return 0;

}



**Q7. Write a program to make use of inline function.**

#include "iostream"

using namespace std;

inline int sum(int a,int b)

{

return (a+b);

}

int main()

{

int a,b;

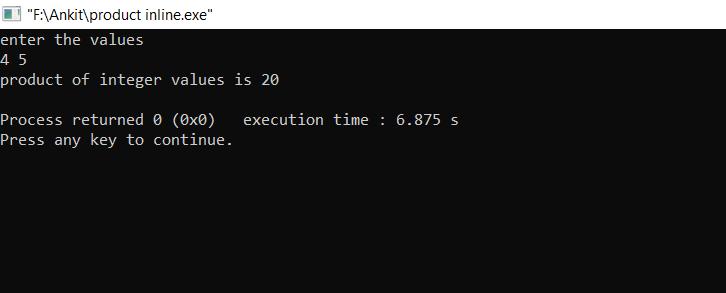
cout<<"Enter two numbers : ";

cin>>a>>b;

cout<<"\nSum is : "<<sum(a,b);

return 0;

}

****

**Q8. Write a program to input two numbers from the user, make an inline function to calculate product of these numbers and display the results.**

#include "iostream"

using namespace std;

inline void product(int p,int q)

{

cout<<"product of integer values is "<<p\*q<<"\n";

}

int main()

{

int a,b;

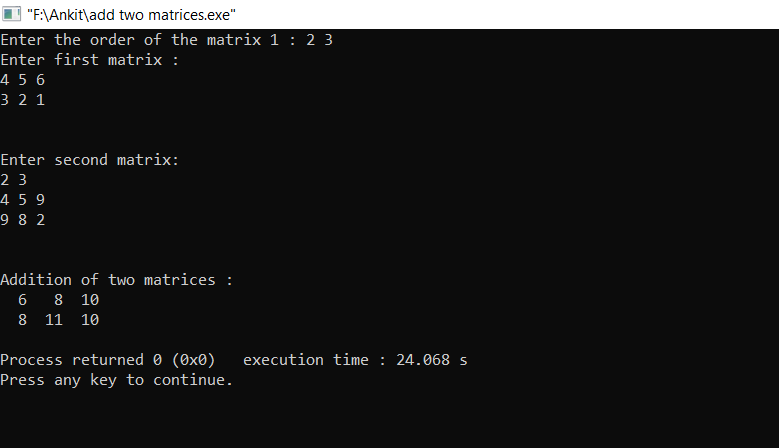
cout<<"enter the values\n";

cin>>a>>b;

product(a,b);

return 0;

}



**Q9.** **Write a program to add two matrices.**

#include "iostream"

using namespace std;

int main()

{

int p,q,i,j,k;

cout<<"Enter the order of the matrix 1 : ";

cin>>p>>q;

int a[p][q],b[p][q],c[p][q];

cout<<"Enter first matrix : \n";

for(i=0;i<p;i++)

for(j=0;j<q;j++)

cin>>a[i][j];

cout<<endl;

cout<<"\nEnter second matrix:\n";

for(i=0;i<p;i++)

for(j=0;j<q;j++)

{

cin>>b[i][j];

c[i][j]=a[i][j]+b[i][j];

}

cout<<endl;

cout<<"\nAddition of two matrices : \n";

for(i=0;i<p;i++)

{

for(j=0;j<q;j++)

{

cout.width(3);

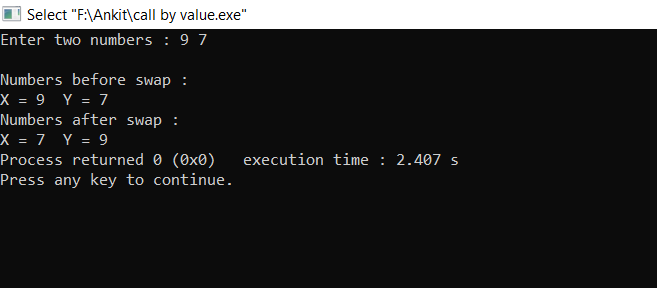
cout<<c[i][j]<<" ";

}

cout<<endl;

}

}



**Q10. Write a program to implement swapping function using call by value.**

#include "iostream"

using namespace std;

void swap(int x, int y)

{

int z;

z=x;

x=y;

y=z;

cout<<"\nNumbers after swap :"<<endl;

cout<<"X = "<<x<<" Y = "<<y;

}

int main()

{

int a,b;

cout<<"Enter two numbers : ";

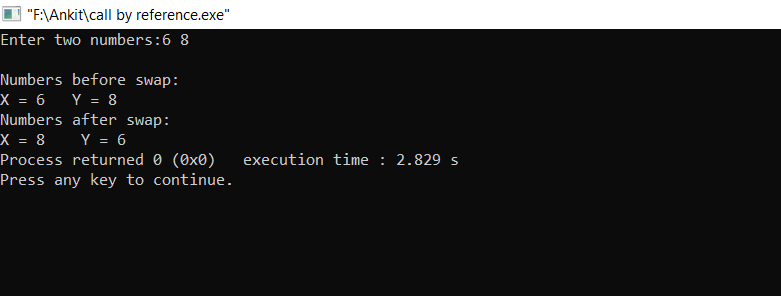
cin>>a>>b;

cout<<"\nNumbers before swap : "<<endl;

cout<<"X = "<<a<<" Y = "<<b;

swap(a,b);

}



Q11. **Write a program to implement swapping function using call by reference.**

#include "iostream"

using namespace std;

void swap(int \*x,int \*y)

{

int z;

z=\*x;

\*x=\*y;

\*y=z;

}

int main()

{

int a,b;

cout<<"Enter two numbers:";

cin>>a>>b;

cout<<"\nNumbers before swap:\n";

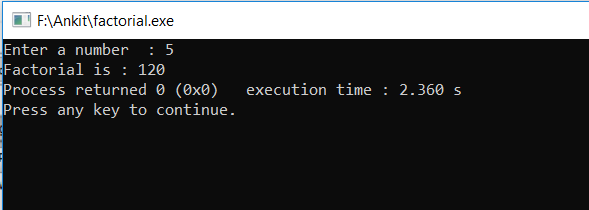
cout<<"X = "<<a<<" Y = "<<b;

swap(&a,&b);

cout<<"\nNumbers after swap:\n";

cout<<"X = "<<a<<" Y = "<<b;

}



**Q12. Write a program to print the factorial of a number using recursion.**

#include "iostream"

using namespace std;

int fact(int n)

{

if(n==0)

return 1;

else

return (n\*fact(n-1));

}

int main()

{

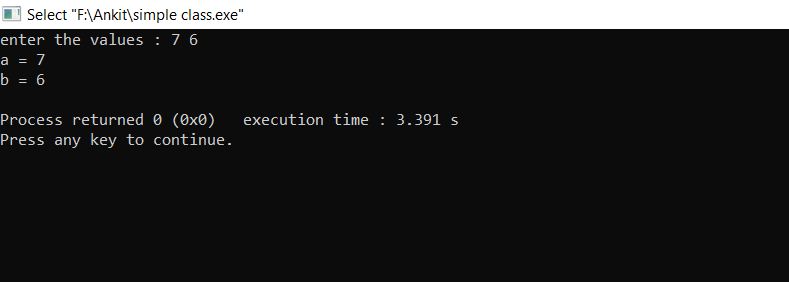
int n;

cout<<"Enter a number : ";

cin>>n;

cout<<"Factorial is : "<<fact(n);

}



Q13. **Create a simple class containing two data items in private and two member functions in public. Make an object of this class and call these two functions.**

#include "iostream"

using namespace std;

class abc

{

int a;

int b;

public:

void getdata(int x, int y)

{

a=x;

b=y;

}

void display()

{

cout<<"a = "<<a<<"\n";

cout<<"b = "<<b<<"\n";

}

};

int main()

{

abc object;

int x,y;

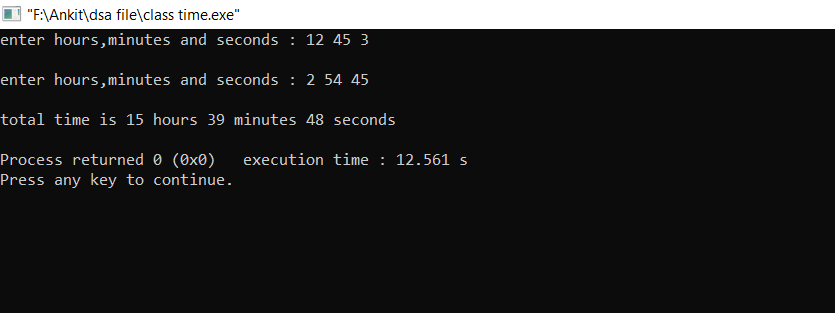
cout<<"enter the values : ";

cin>>x>>y;

object.getdata(x,y);

object.display();

}



**Q14.** **Write a program to read and add two times using class.**

#include"iostream"

using namespace std;

class time

{

int h=0,m=0,s=0;

public:

void settime()

{

cout<<"enter hours,minutes and seconds : ";

cin>>h>>m>>s;

cout<<endl;

}

void addtime(time x,time y)

{

s=x.s+y.s;

m=x.m+y.m;

h=x.h+y.h;

if(s>=60)

{

s=s%60;

m++;

}

if(m>=60)

{

m=m%60;

h++;

}

cout<<"total time is "<<h<<" hours "<<m<<" minutes "<<s<<" seconds \n";

}

};

int main()

{

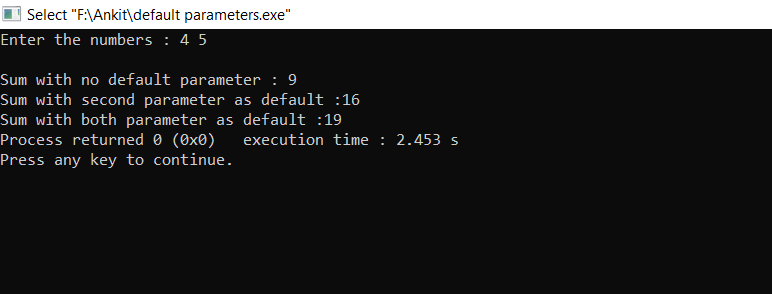
time t1,t2,t3;

t1.settime();

t2.settime();

t3.addtime(t1,t2);

}



**Q15. Write a program to implement default parameters where you declare a function with default parameters and you call the function**

**(i). With no default arguments**

**(ii). With 2nd parameter as defaulted**

**(iii). With both parameters as defaulted**

#include "iostream"

using namespace std;

int sum(int p=7 , int q=12)

{

return (p+q);

}

int main()

{

int a,b;

cout<<"Enter the numbers : ";

cin>>a>>b;

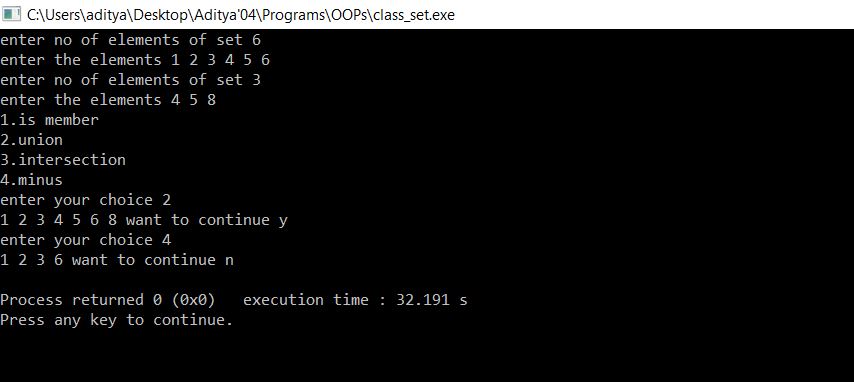
cout<<"\nSum with no default parameter : "<<sum(a,b);

cout<<"\nSum with second parameter as default :"<<sum(a);

cout<<”\nSum with both parameter as default :"<<sum();

return 0;

}



**Q16. Create class set to store the integer values and implement following member functions**

**(i). Is\_member**

**(ii). Union**

**(iii). intersection**

**(iv). minus**

#include"iostream"

using namespace std;

class set

{

int a[10],len;

public:

void create\_set();

void is\_member();

void set\_union(set,set);

void set\_intersection(set,set);

void set\_minus(set,set);

void display();

};

void set::create\_set()

{ int i;

cout<<"enter no of elements of set ";

cin>>len;

cout<<"enter the elements ";

for(i=0;i<len;i++)

cin>>a[i];

}

void set::is\_member()

{ int d,i;

cout<<"enter the no you want to search ";

cin>>d;

for(i=0;i<len;i++)

{

if(a[i]==d)

break;

}

if(i==len)

cout<<"not present";

else

cout<<"present";

}

void set::set\_union(set x,set y)

{

int i,j,k=0;

for(i=0;i<x.len;i++)

a[k++]=x.a[i];

for(i=0;i<y.len;i++)

{

for(j=0;j<x.len;j++)

{

if(y.a[i]==x.a[j])

break;

}

if(j==x.len)

a[k++]=y.a[i];

}

len=k;

}

void set::set\_intersection(set x,set y)

{

int i,j,k=0;

for(i=0;i<x.len;i++)

{

for(j=0;j<y.len;j++)

{

if(x.a[i]==y.a[j])

break;

}

if(j<y.len)

a[k++]=x.a[i];

}

len=k;

}

void set::set\_minus(set x,set y)

{

int i,j,k=0;

for(i=0;i<x.len;i++)

{

for(j=0;j<y.len;j++)

{

if(x.a[i]==y.a[j])

break;

}

if(j==y.len)

a[k++]=x.a[i];

}

len=k;

}

void set::display()

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

cout<<a[i]<<" ";

}

int main()

{ set A,B,C,D,E;

A.create\_set();

B.create\_set();

cout<<"1.is member\n"

<<"2.union\n"

<<"3.intersection\n"

<<"4.minus\n";

int choice;

char z='y';

while(z=='y'){

cout<<"enter your choice ";

cin>>choice;

switch(choice)

{

case 1:

A.is\_member();

break;

case 2:

C.set\_union(A,B);

C.display();

break;

case 3:

D.set\_intersection(A,B);

D.display();

break;

case 4:

E.set\_minus(A,B);

E.display();

}

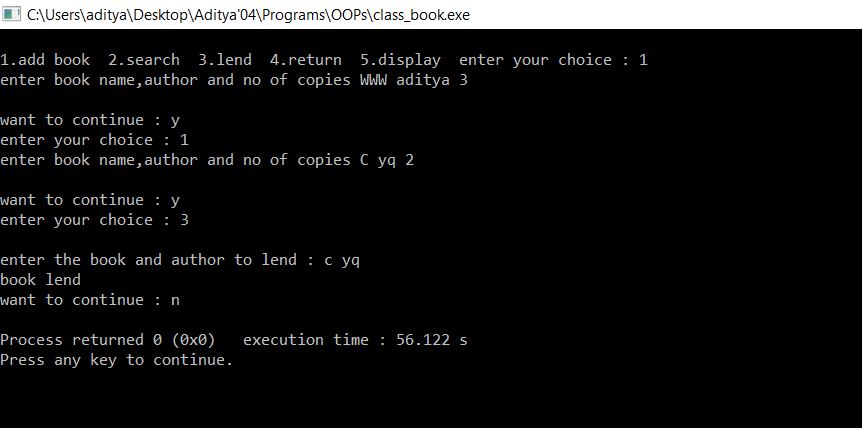
cout<<"want to continue ";

z=getche();

}

return 0;

}



**Q17. Create a class book to store details of book like book name, author name, and copies count. Create 20 different books.**

#include "iostream"

#include ”cstring”

using namespace std;

int a;

class book

{

char bname[10],aname[10];

int count;

public:

void add();

friend int search(book\*,char\*,char\*);

void lend();

void return1();

friend void display(book\*);

};

void book::add()

{

cout<<"enter book name,author and no of copies ";

cin>>bname>>aname>>count;

a++;

}

int search(book\*c,char\*s,char\*t)

{

for(int i=0;i<a;i++)

{

if(strcmp(c[i].bname,s)==0 && strcmp(c[i].aname,t)==0)

return i;

}

return -1;

}

void book::lend()

{

if(count>1)

{

cout<<"book lend";

count-=1;

}

else

cout<<"not sufficient copies to lend";

}

void book::return1()

{

cout<<"book returned";

count+=1;

}

void display(book \*c)

{

for(int i=0;i<a;i++)

cout<<c[i].bname<<" "<<c[i].aname<<" "<<c[i].count<<"\n";

}

int main()

{

book b[20];

int i=0,f;

char d[10],e[10];

cout<<"1.add book\n"

<<"2.search\n"

<<"3.lend\n"

<<"4.return\n"

<<"5.display\n";

char z='y';

int choice;

while(z=='y')

{ cout<<"enter your choice";

cin>>choice;

switch(choice)

{

case 1:

b[i].add();

i++;

break;

case 2:

cout<<"enter the book and author you want to search ";

cin>>d>>e;

f=search(b,d,e);

if(f==-1)

cout<<"book is unavailable";

else

cout<<"book is available";

break;

case 3:

cout<<"enter the book and author to lend";

cin>>d>>e;

f=search(b,d,e);

b[f].lend();

break;

case 4:

cout<<"enter the book and author to return";

cin>>d>>e;

f=search(b,d,e);

b[f].return1();

break;

case 5:

display(b);

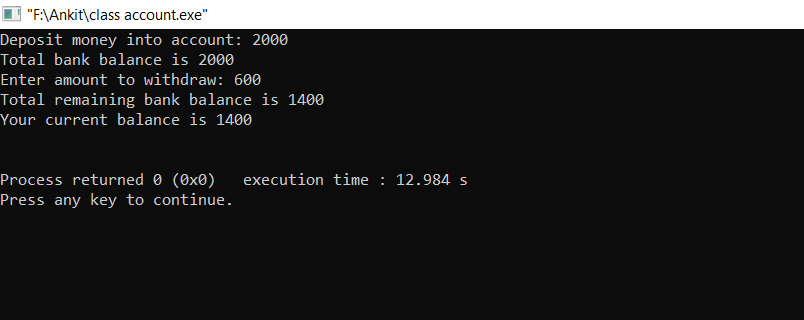
}

cout<<"want to continue";

z=getche();

}

}



**Q18. Make a class account that represents your bank account. This class should allow you to deposit and withdraw money and show the account details.**

#include "iostream"

using namespace std;

class account

{

int a;

public:

void deposit(int b)

{

a=0;

a=a+b;

cout<<"Total bank balance is "<<a<<"\n";

}

void withdraw(int b)

{

if(a<b)

cout<<"Not enough balance\n";

else

{

a=a-b;

cout<<"Total remaining bank balance is "<<a<<"\n";

}

}

void display()

{

cout<<"Your current balance is "<<a<<"\n\n";

}

};

int main()

{

int w,b;

account acc;

cout << "Deposit money into account: ";

cin >>b;

acc.deposit(b);

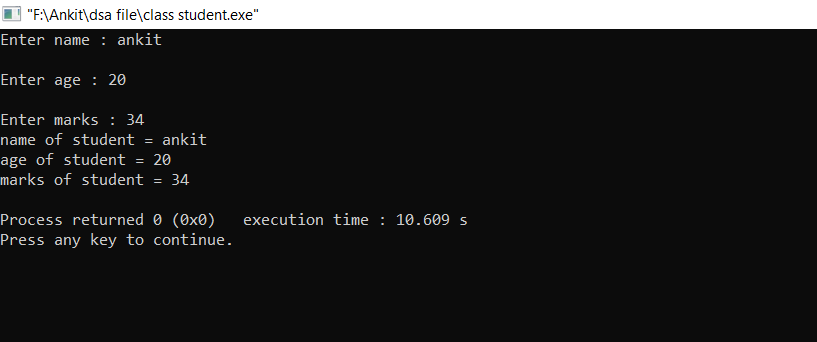
cout << "Enter amount to withdraw: ";

cin >> w;

acc.withdraw(w);

acc.display();

}



**Q19. Create a class student to get the details of the students and print them.**

#include "iostream"

using namespace std;

class student

{

char name[100];

int age;

int marks;

public:

void setdata()

{

cout<<"Enter name : ";

cin>>name;

cout<<"\nEnter age : ";

cin>>age;

cout<<"\nEnter marks : ";

cin>>marks;

}

void display()

{

cout<<"name of student = "<<name<<"\nage of student = "<<age<<"\nmarks of student = "<<marks<<"\n";

}

};

int main()

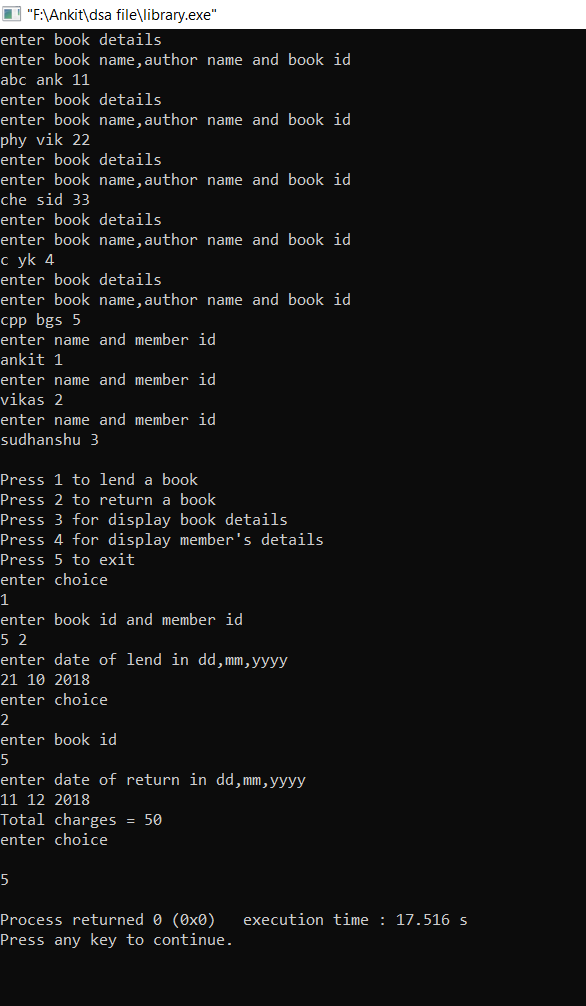
{

student a;

a.setdata();

a.display();

}



**Q20. Write a program to create a library that can lend the book and add important details.**

#include<iostream>

#include<string.h>

using namespace std;

class member;

struct date

{

int dd,mm,yyyy;

};

class book

{

char book\_name[20],author\_name[20];

static int c;

int bookid,n,memberid;

date dol,dor;

public:

book()

{

n=1;

memberid=-1;

cout<<"enter book details\n";

cout<<"enter book name,author name and book id\n";

cin>>book\_name>>author\_name>>bookid;

}

friend void lend(book&,member&);

friend void return\_book(book&);

int search\_book(int a)

{

if(bookid==a)

return 1;

else

return 0;

}

void display()

{

cout<<"\nBook name="<<book\_name<<"\nAuthor="<<author\_name<<"\nBook id="<<bookid<<"\n";

cout<<"remaining copies = "<<n<<"\n";

cout<<"total number of issued books = "<<c<<"\n";

cout<<"Member id = "<<memberid<<"\n";

}

} ;

class member

{

int id;

char name[20];

public:

member()

{

cout<<"enter name and member id\n";

cin>>name>>id;

}

friend void lend(book&,member&);

int search\_member(int a)

{

if(id==a)

return 1;

else

return 0;

}

void display()

{

cout<<"\nName = "<<name<<"\nMember id = "<<id<<"\n";

}

};

int book::c=0;

void lend(book &a,member &b)

{

if(a.n==1)

{

a.memberid=b.id;

cout<<"enter date of lend in dd,mm,yyyy\n";

cin>>a.dol.dd>>a.dol.mm>>a.dol.yyyy;

a.n=0;

a.c+=1;

}

else

cout<<"Book is not available\n";

}

void return\_book(book &a)

{

a.n=1;

cout<<"enter date of return in dd,mm,yyyy\n";

cin>>a.dor.dd>>a.dor.mm>>a.dor.yyyy;

int x,y,z;

z=a.dor.yyyy-a.dol.yyyy;

y=(a.dor.mm-a.dol.mm)+z\*12;

x=(a.dor.dd-a.dol.dd)+y\*30;

cout<<"Total charges = "<<x<<"\n";

a.c-=1;

a.memberid=-1;

}

int main()

{

book a[5];

member b[3];

cout<<"\nPress 1 to lend a book\nPress 2 to return a book\nPress 3 for display book details\n";

cout<<"Press 4 for display member's details\nPress 5 to exit\n";

cout<<"enter choice\n";

int y;

int i,j,k,l,m;

cin>>y;

while(y!=5)

{

switch(y)

{

case 1:

cout<<"enter book id and member id\n";

cin>>i>>j;

k=0;

while(k<=4)

{

l=a[k].search\_book(i);

if(l==1)

break;

k++;

}

m=0;

while(m<=2)

{

l=b[m].search\_member(j);

if(l==1)

break;

m++;

}

lend(a[k],b[m]);

break;

case 2:

cout<<"enter book id\n";

cin>>i;

k=0;

while(k<=4)

{

l=a[k].search\_book(i);

if(l==1)

break;

k++;

}

return\_book(a[k]);

break;

case 3:

k=1;

while(k!=6)

{

a[k-1].display();

k++;

}

break;

case 4:

k=1;

while(k!=4)

{

b[k-1].display();

k++;

}

break;

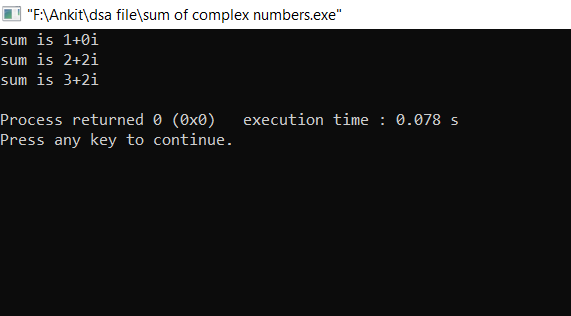
}

cout<<"enter choice\n";

cin>>y;

}

}



**Q21. Write a program to take input for sum of complex number by using default constructor.**

#include<iostream>

using namespace std;

class complex1

{

int i,j;

public:

complex1()

{

i=0;j=0;

}

complex1(int x)

{

i=x;j=0;

}

complex1(int x,int y)

{

i=x;j=y;

}

void sum(complex1 a)

{

int x,y;

x=i+a.i;

y=j+a.j;

cout<<"sum is "<<x<<"+"<<y<<"i\n";

}

void display()

{

cout<<i<<"+"<<j<<"i\n";

}

};

int main()

{

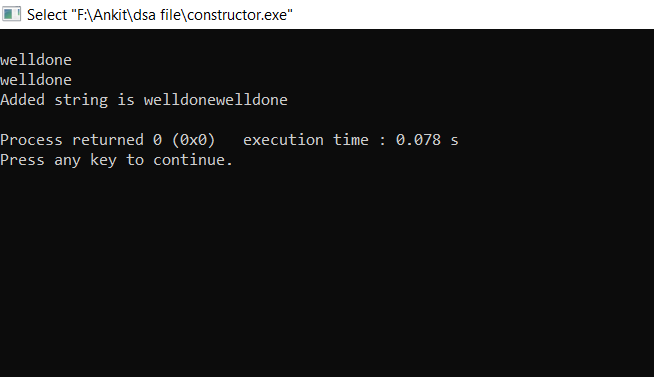
complex1 a,b(1),c(2,2);

a.sum(b);

a.sum(c);

b.sum(c);

}



**Q22. Write a program to class string to perform basic functions using constructors.**

#include<iostream>

#include<string.h>

using namespace std;

class string1

{

char a[100];

public:

string1()

{

a[0]='\0';

}

string1(char b[])

{

strcpy(a,b);

}

void addstring(string1 x,string1 y)

{

int i=0,j=0;

while(x.a[i]!='\0')

{

a[i]=x.a[i];

i++;

}

while(1)

{

a[i]=y.a[j];

i++;

j++;

if(y.a[j]=='\0')

{

a[i]='\0';

break;

}

}

cout<<"Added string is "<<a<<"\n";

}

void display()

{

cout<<a<<"\n";

}

};

int main()

{

char s1[20]="'\0'",s2[20]="welldone";

string1 c,d(s2),e(s2);

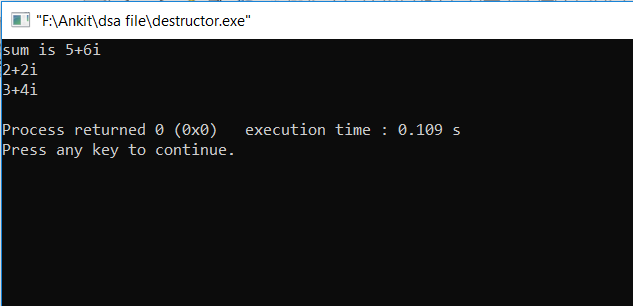
c.display();

d.display();

e.display();

c.addstring(d,e);

}



**Q23. Write a program to demonstrate concept of destructor**.

#include<iostream>

using namespace std;

class complex1

{

int i,j;

public:

complex1(int x,int y)

{

i=x;j=y;

}

void sum(complex1 &a)

{

int x,y;

x=i+a.i;

y=j+a.j;

cout<<"sum is "<<x<<"+"<<y<<"i\n";

}

void display()

{

cout<<i<<"+"<<j<<"i\n";

}

~complex1()

{

display();

}

};

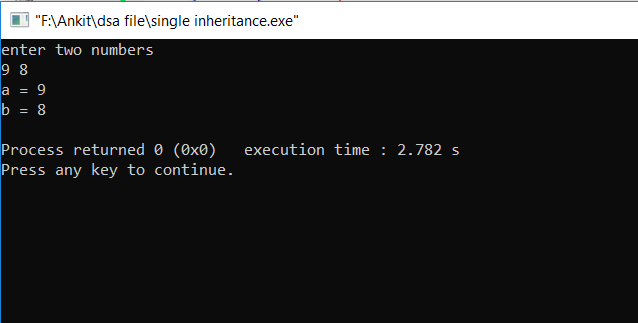
int main()

{

complex1 a(3,4),c(2,2);

a.sum(c);

}



**Q24. Write a program to demonstrate single inheritance**

#include<iostream>

using namespace std;

class A

{

protected:

int a;

};

class B:public A

{

protected:

int b;

public:

void get\_ab(int x,int y)

{

a=x;

b=y;

}

void display()

{

cout<<"a = "<<a<<"\nb = "<<b<<"\n";

}

};

int main()

{

int x,y;

cout<<"enter two numbers\n";

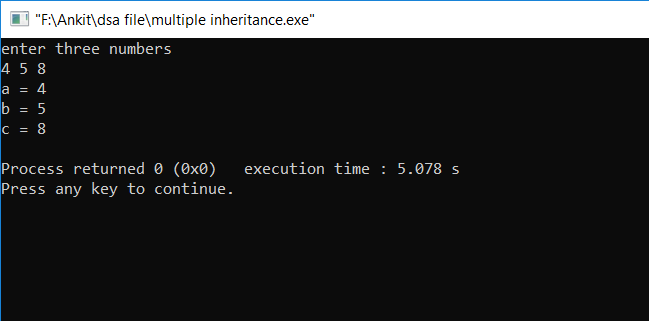
cin>>x>>y;

B obj2;

obj2.get\_ab(x,y);

obj2.display();

}



**Q25. Write a program to demonstrate multiple inheritance**.

#include<iostream>

using namespace std;

class A

{

protected:

int a;

public:

void display\_a()

cout<<"a = "<<a<<"\n";

};

class B

{

protected:

int b;

public:

void display\_b()

cout<<"b = "<<b<<"\n";

};

class C:public A,public B

{

protected:

int c;

public:

void get\_abc(int x,int y,int z)

{

a=x;

b=y;

c=z;

}

void display\_c()

cout<<"c = "<<c<<"\n";

};

int main()

{

int x,y,z;

cout<<"enter three numbers\n";

cin>>x>>y>>z;

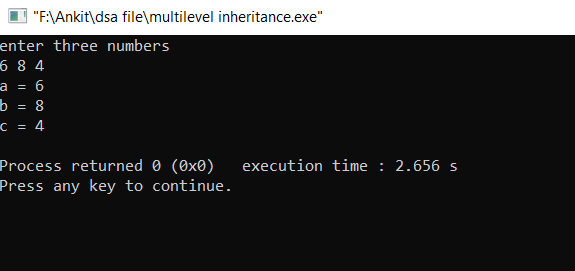
C obj3;

obj3.get\_abc(x,y,z);

obj3.display\_a();

obj3.display\_b();

obj3.display\_c(); }



**Q26. Write a program to demonstrate multilevel inheritance.**

#include<iostream>

using namespace std;

class A

{

protected:

int a;

};

class B:public A

{

protected:

int b;

};

class C:public B

{

protected:

int c;

public:

void get\_abc(int x,int y,int z)

{

a=x;

b=y;

c=z;

}

void display()

{

cout<<"a = "<<a<<"\nb = "<<b<<"\nc = "<<c<<"\n";

}

};

int main()

{

int x,y,z;

cout<<"enter three numbers\n";

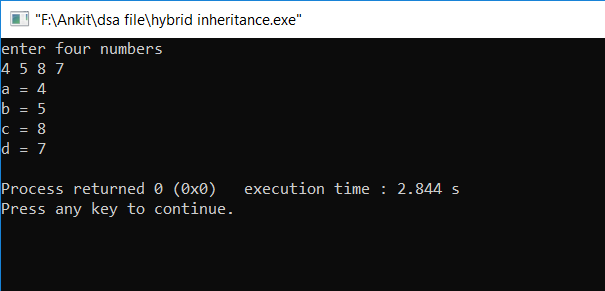
cin>>x>>y>>z;

C obj3;

obj3.get\_abc(x,y,z);

obj3.display();

}



**Q27. Write a program to demonstrate hybrid inheritance.**

#include<iostream>

using namespace std;

class A

{

protected:

int a;

};

class B:public A

{

protected:

int b;

};

class D

{

protected:

int d;

};

class C:public B,public D

{

protected:

int c;

public:

void get\_abcd(int x,int y,int z,int k)

{

a=x;

b=y;

c=z;

d=k;

}

void display()

{

cout<<"a = "<<a<<"\nb = "<<b<<"\nc = "<<c<<"\nd = "<<d<<"\n";

}

};

int main()

{

int x,y,z,k;

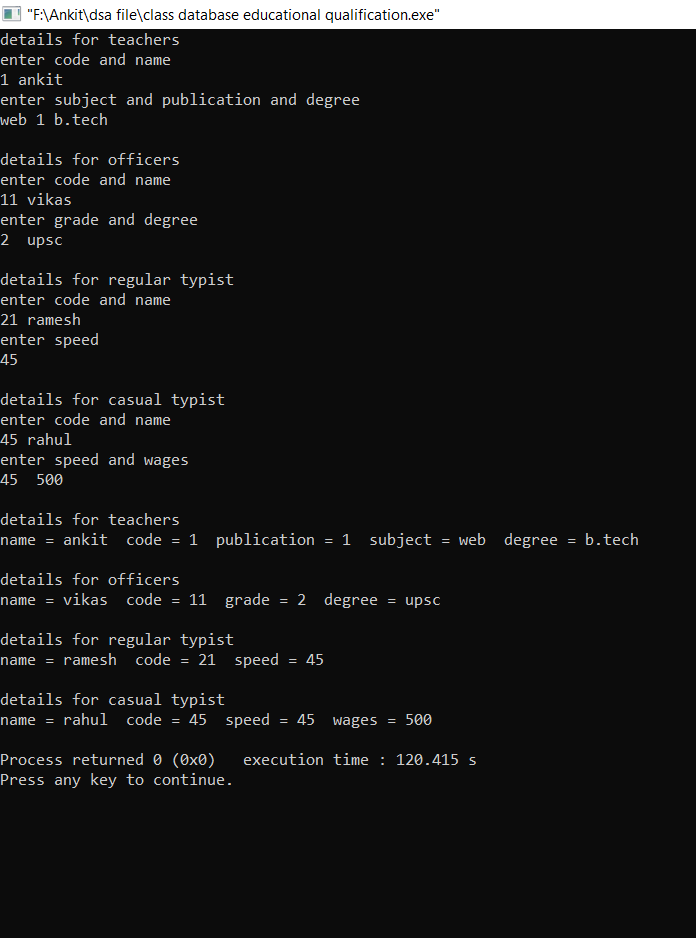
cout<<"enter four numbers\n";

cin>>x>>y>>z>>k;

C obj3;

obj3.get\_abcd(x,y,z,k);

obj3.display(); }



**Q28. Write a program to improve the last program and increase the educational qualification class.**

#include<iostream>

#include<string.h>

using namespace std;

class staff

{

protected:

int code;

char name[20];

};

class teacher:public staff

{

protected:

int publication;

char subject[20];

};

class typist:public staff

{

protected:

int speed;

};

class officer:public staff

{

protected:

char grade;

};

class education:public teacher,public officer

{

protected:

char degree1[20];

char degree2[20];

public:

void getdata\_t(int c,char n[],int p,char s[],char d[])

{

teacher::code=c;

strcpy(teacher::name,n);

publication=p;

strcpy(subject,s);

strcpy(degree1,d);

}

void getdata\_o(int c,char n[],char g,char d[])

{

officer::code=c;

strcpy(officer::name,n);

grade=g;

strcpy(degree2,d);

}

void display\_t()

{

cout<<"\ndetails for teachers";

cout<<"\nname = "<<teacher::name<<" code = "<<teacher::code;

cout<<" publication = "<<publication<<" subject = "<<subject;

cout<<" degree = "<<degree1<<"\n";

}

void display\_o()

{

cout<<"\ndetails for officers";

cout<<"\nname = "<<officer::name<<" code = "<<officer::code;

cout<<" grade = "<<grade;

cout<<" degree = "<<degree2<<"\n";

} };

class regular:public typist

{

public:

void getdata(int c,char n[],int sp)

{

code=c;

strcpy(name,n);

speed=sp;

}

void display()

{

cout<<"\ndetails for regular typist";

cout<<"\nname = "<<name<<" code = "<<code<<" speed = "<<speed<<"\n";

} };

class casual:public typist

{

protected:

int wages;

public:

void getdata(int c,char n[],int sp,int w)

{

code=c;

strcpy(name,n);

speed=sp;

wages=w;

}

void display()

{

cout<<"\ndetails for casual typist";

cout<<"\nname = "<<name<<" code = "<<code<<" speed = "<<speed<<" wages = "<<wages<<"\n";

}};

int main()

{

education e;

regular r;

casual c;

int a,b,d;

char s[20],n[20],g,x[20];

cout<<"details for teachers\n";

cout<<"enter code and name\n";

cin>>a>>n;

cout<<"enter subject and publication and degree\n";

cin>>s>>b>>x;

e.getdata\_t(a,n,b,s,x);

cout<<"\ndetails for officers\n";

cout<<"enter code and name\n";

cin>>a>>n;

cout<<"enter grade and degree\n";

cin>>g>>x;

e.getdata\_o(a,n,g,x);

cout<<"\ndetails for regular typist\n";

cout<<"enter code and name\n";

cin>>a>>n;

cout<<"enter speed\n";

cin>>b;

r.getdata(a,n,b);

cout<<"\ndetails for casual typist\n";

cout<<"enter code and name\n";

cin>>a>>n;

cout<<"enter speed and wages\n";

cin>>b>>d;

c.getdata(a,n,b,d);

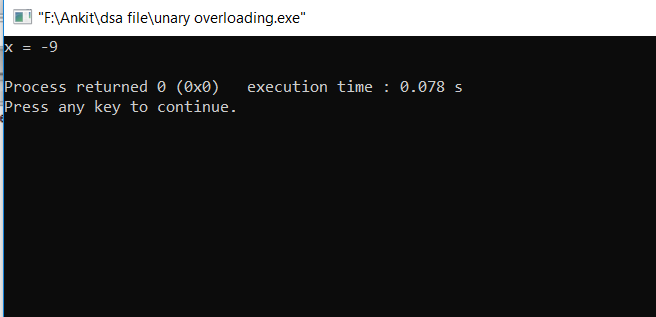
e.display\_t();

e.display\_o();

r.display();

c.display();

}



**Q29. Write a program to show overloading.**

**(i). Unary overloading**

#include<iostream>

using namespace std;

class a;

class a

{

int x;

public:

a(int b)

{

x=b;

}

a(){}

void display()

{

cout<<"x = "<<x<<"\n";

}

a operator-()

{

a t;

t.x=-x;

return t;

}

};

int main()

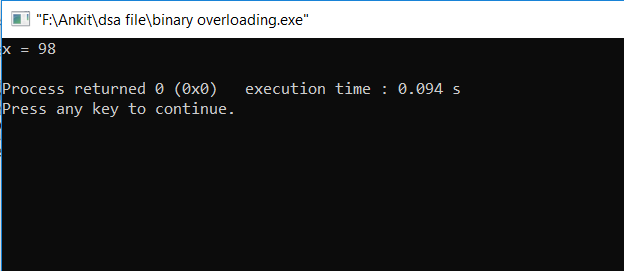
{

a obj1=4,obj2;

obj2=-obj1;

obj2.display();

}



**(ii).binary overloading.**

#include<iostream>

using namespace std;

class a;

class a

{

int x;

public:

a(int b)

{

x=b;

}

a(){}

void display()

{

cout<<"x = "<<x<<"\n";

}

a operator+(a temp)

{

a t;

t.x=x+temp.x;

return t;

}

};

int main()

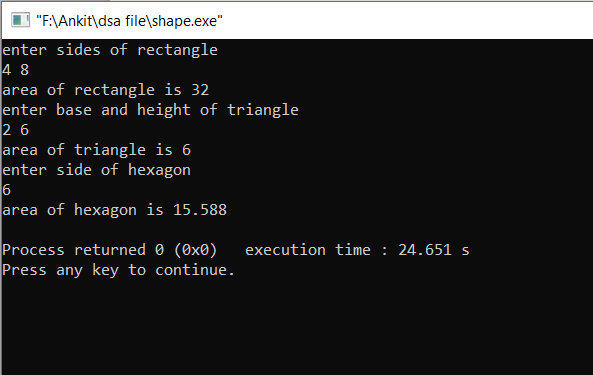
{

a obj1=4,obj2=5,obj3;

obj3=obj1+obj2;

obj3.display();

}



**Q30. Write a program to create a class shape to calculate area of rectangle, triangle and hexagon using inheritance.**

#include<iostream>

#include<string.h>

using namespace std;

class shape

{

protected:

float area;

public:

float get\_area();

};

class rectangle:public shape

{

int a;

int b;

public:

get\_ab(int x,int y)

{

a=x;

b=y;

}

float get\_area()

{

area=a\*b;

return area;

}

};

class triangle:public shape

{

float b;

float h;

public:

get\_bh(float x,float y)

{

b=x;

h=y;

}

float get\_area()

{

area=(h\*b)/2;

return area;

}

};

class hexagon:public shape

{

float a;

public:

get\_a(float x)

{

a=x;

}

float get\_area()

{

area=(1.732\*a\*a)/4;

return area;

}

};

int main()

{

rectangle r;

triangle t;

hexagon h;

int a,b;

float x,y,z;

cout<<"enter sides of rectangle\n";

cin>>a>>b;

r.get\_ab(a,b);

z=r.get\_area();

cout<<"area of rectangle is "<<z<<"\n";

cout<<"enter base and height of triangle\n";

cin>>x>>y;

t.get\_bh(x,y);

z=t.get\_area();

cout<<"area of triangle is "<<z<<"\n";

cout<<"enter side of hexagon\n";

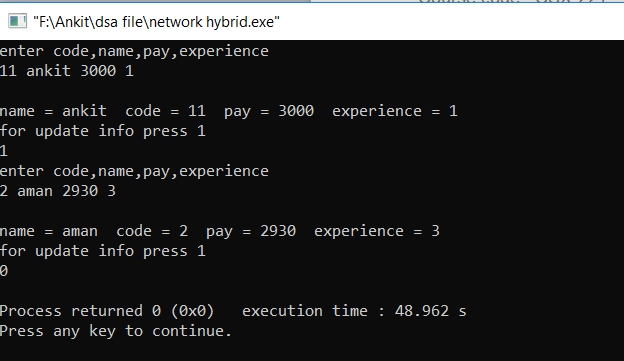
cin>>a;

h.get\_a(a);

z=h.get\_area();

cout<<"area of hexagon is "<<z<<"\n";

}



**Q31. Write a program to create network of the classes (hybrid inheritance) by classes person, account, admin and master.**

#include<iostream>

#include<string.h>

using namespace std;

class person

{

protected:

int code;

char name[20];

};

class account:public person

{

protected:

int pay;

};

class admin:public person

{

protected:

int experience;

};

class master:public admin,public account

{

public:

void getdata(int c,char n[],int p,int exp)

{

admin::code=c;

strcpy(admin::name,n);

pay=p;

experience=exp;

}

void display()

{

cout<<"\nname = "<<admin::name<<" code = "<<admin::code<<" pay = "<<pay<<" experience = "<<experience<<"\n";

}

};

int main()

{

master m;

int c,p,e;

char n[20];

abc: cout<<"enter code,name,pay,experience\n";

cin>>c>>n>>p>>e;

m.getdata(c,n,p,e);

m.display();

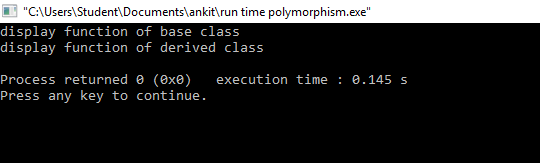
cout<<"for update info press 1\n";

cin>>e;

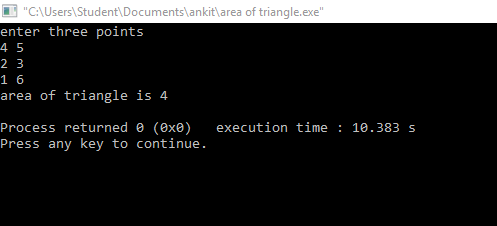
if(e)

goto abc;

}



**Q32. write a program to show the concept of run-time polymorphism using virtual function.**

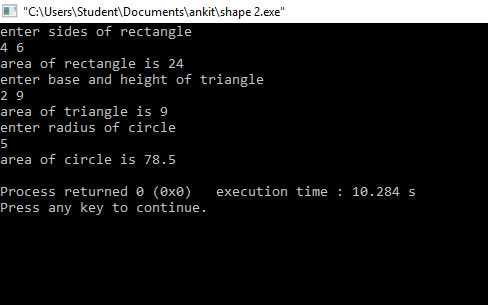
#include<iostream>  
using namespace std;  
class base  
{  
public:  
 virtual void display()  
 {  
 cout<<"display function of base class\n";  
 }  
};  
class derived:public base  
{  
public:  
 void display()  
 {  
 cout<<"display function of derived class\n";  
 }  
};  
  
int main()  
{  
 base \*p,obj;  
 derived obj1;  
 p=&obj;  
 p->display();  
 p=&obj1;  
 p->display();  
}

**Q33. Write a program to find area of triangle by three different points using containership.**

#include<iostream>  
using namespace std;  
class G  
{  
 int g;  
public:  
 void get(int a)  
 {  
 g=a;  
 }  
 void display()  
 {  
 cout<<"g = "<<g<<"\n";  
 }  
};  
class B  
{  
 int b;  
public:  
 void get(int a)  
 {  
 b=a;  
 }  
 void display()  
 {  
 cout<<"b = "<<b<<"\n";  
 }  
};  
class E  
{  
 int e;  
public:  
 void display(int a,int b,int c)  
 {  
 G obj1;  
 B obj2;  
 obj1.get(a);  
 obj2.get(b);  
 e=c;  
 cout<<"e = "<<e<<"\n";

obj1.display();

obj2.display();  
 }  
};  
  
int main()  
{  
 int x,y,z,k;  
 E obj;  
 cout<<"enter three numbers\n";  
 cin>>x>>y>>z;  
 obj.display(x,y,z);  
}



**Q34. Improvise the above SHAPE program to display the area of circle.**

#include<iostream>

using namespace std;

class shape

{

protected:

float a,b;

public:

void get\_data(float x,float y=0)

{

a=x;

b=y;

}

void display\_area()

{}

};

class rectangle:public shape

{

public:

void display\_area()

{

cout<<"area of rectangle is "<<a\*b<<"\n";

}

};

class triangle:public shape

{

public:

void display\_area()

{

cout<<"area of triangle is "<<(a\*b)/2<<"\n";

}

};

class circle : public shape

{

public:

void display\_area()

{

cout<<"area of circle is "<<3.14\*(a\*a)<<"\n";

}

};

int main()

{

rectangle r;

triangle t;

circle c;

float x,y;

cout<<"enter sides of rectangle\n";

cin>>x>>y;

r.get\_data(x,y);

r.display\_area();

cout<<"enter base and height of triangle\n";

cin>>x>>y;

t.get\_data(x,y);

t.display\_area();

cout<<"enter radius of circle\n";

cin>>x;

c.get\_data(x);

c.display\_area();

return 0;

}