**Session 7 (unit-4): Constructors & Destructors and Error handling**

1.      WAP that accepts and adds two complex numbers using :

1. Default constructors
2. Parameterized constructors

#include<iostream>

using namespace std;

class addcomplex

{

public:

int real, i;

addcomplex()

{

real = 0;

i = 0;

}

addcomplex(int r, int imaginary)

{

real = r;

i = imaginary;

}

addcomplex add(addcomplex c1, addcomplex c2)

{

addcomplex C;

C.real = c1.real + c2.real;

C.i = c1.i + c2.i;

return C;

}

};

int main()

{

addcomplex c1(8, 5);

cout<<"\nComplex number 1: "<<c1.real<<" + i"<<c1.i<<endl;

addcomplex c2(9, 3);

cout<<"\nComplex number 2: "<<c2.real<<" + i"<<c2.i<<endl;

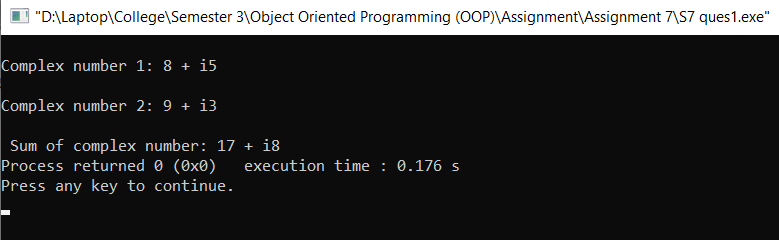
addcomplex C;

C = C.add(c1, c2);

cout<<"\n Sum of complex number: "<<C.real<<" + i"<<C.i;

return 0;

}



1. WAP that calculates the square-root of a number and handles the exception that a number cannot be negative.

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

int a;

int square;

cout<<"Enter the number:";

cin>>a;

if(a>=0)

{

square=sqrt(a);

cout<<"Square root of "<<a<<" is "<<square;

}

else

{

try

{

throw(a);

}

catch(int ex)

{

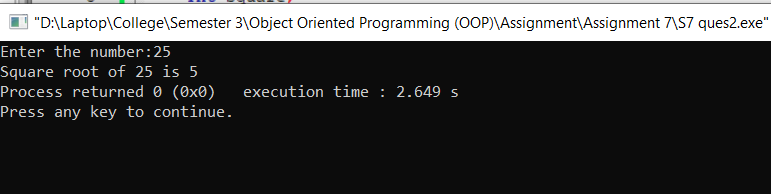
cout<<"number cannot be negative";

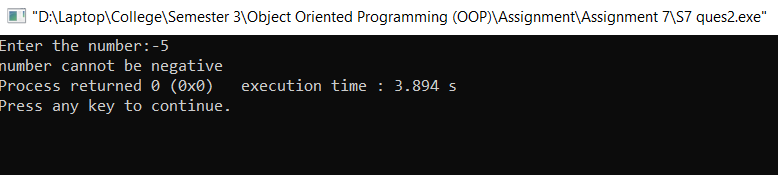
}

}

return 0;

}





1. WAP that checks a person age for voting, all the exceptions are to be handled with exception handling.

#include<iostream>

using namespace std;

int main()

{

int age;

cout<<"Enter your age:";

cin>>age;

if(age>=18)

{

cout<<"Eligible for voting";

}

else

{

try

{

if(age<0)

{

throw(age);

}

else if (age>0 && age<18)

{

throw(age);

}

}

catch(int ex)

{

cout<<"Not eligible for voting";

}

}

return 0;

}

