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

|  |  |
| --- | --- |
| Roll No. A016 | Name: Varun Khadayate |
| Program: B-Tech (CSBS) | Division: SY |
| Batch: 1 | Date of Submission: 21-09-2020 |

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

**ANS:**

**CODE:** #include<iostream>

using namespace std;

class complex

{

public :

int real, img;

};

int main()

{

complex a, b, c;

cout << "Enter a and b where a + ib is the first complex number.";

cout << "\na = ";

cin >> a.real;

cout << "b = ";

cin >> a.img;

cout << "Enter c and d where c + id is the second complex number.";

cout << "\nc = ";

cin >> b.real;

cout << "d = ";

cin >> b.img;

c.real = a.real + b.real;

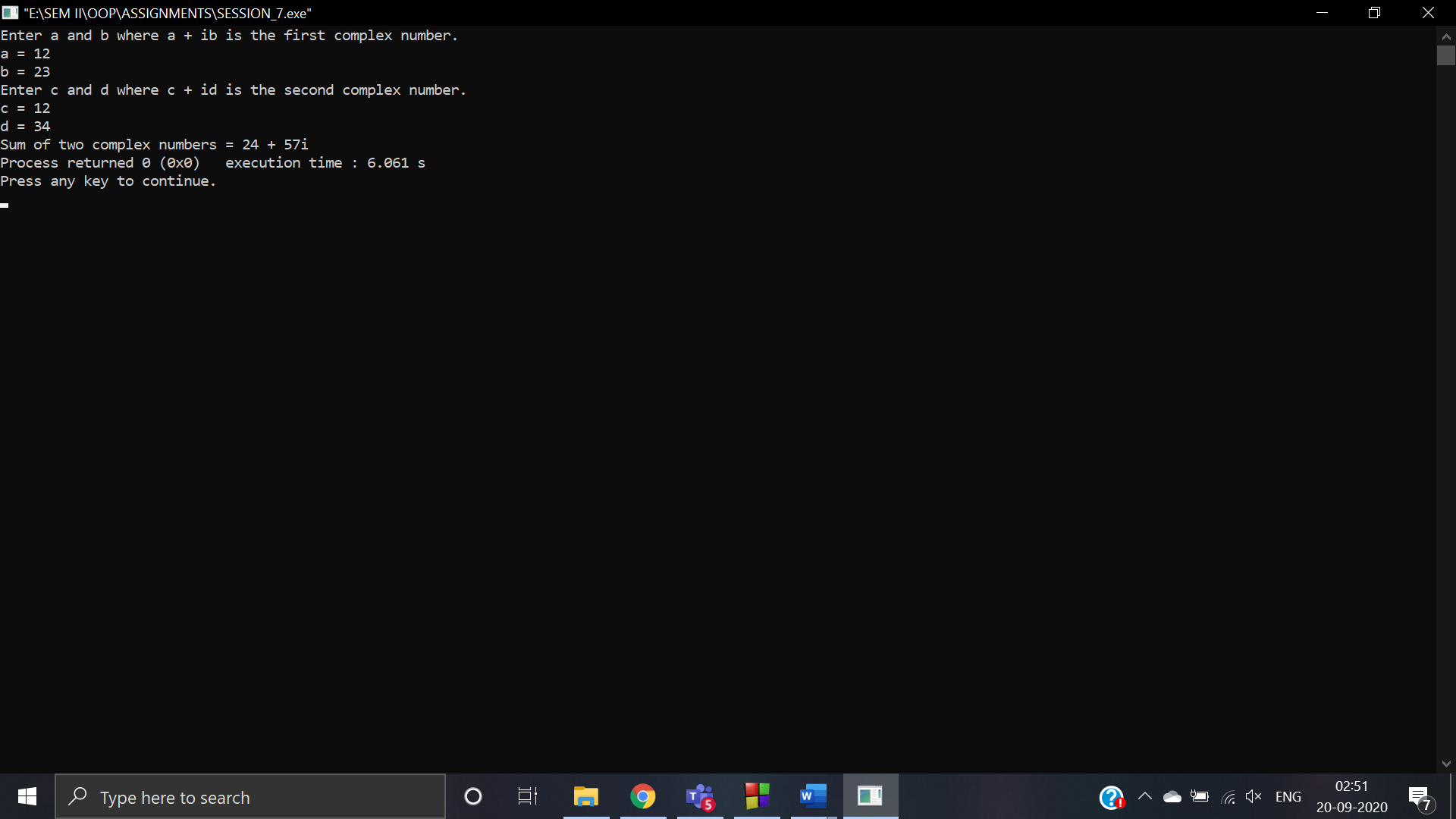
c.img = a.img + b.img;

cout << "Sum of two complex numbers = " << c.real << " + " << c.img << "i";

return 0;

}

**OUTPUT:**



* 1. **Parameterized constructors**

**ANS:**

**CODE:** #include<iostream>

using namespace std;

class complex

{

public:

int real, img;

complex()

{

}

complex(int tempReal, int tempImg)

{

real = tempReal;

img = tempImg;

}

complex addComp(complex a, complex b)

{

complex temp;

temp.real = a.real + b.real;

temp.img = a.img + b.img;

return temp;

}

};

int main()

{

complex a, b, c;

cout << "Enter a and b where a + ib is the first complex number.";

cout << "\na = ";

cin >> a.real;

cout << "b = ";

cin >> a.img;

complex (a.real, a.img);

cout << "Enter c and d where c + id is the second complex number.";

cout << "\nc = ";

cin >> b.real;

cout << "d = ";

cin >> b.img;

complex (b.real, b.img);

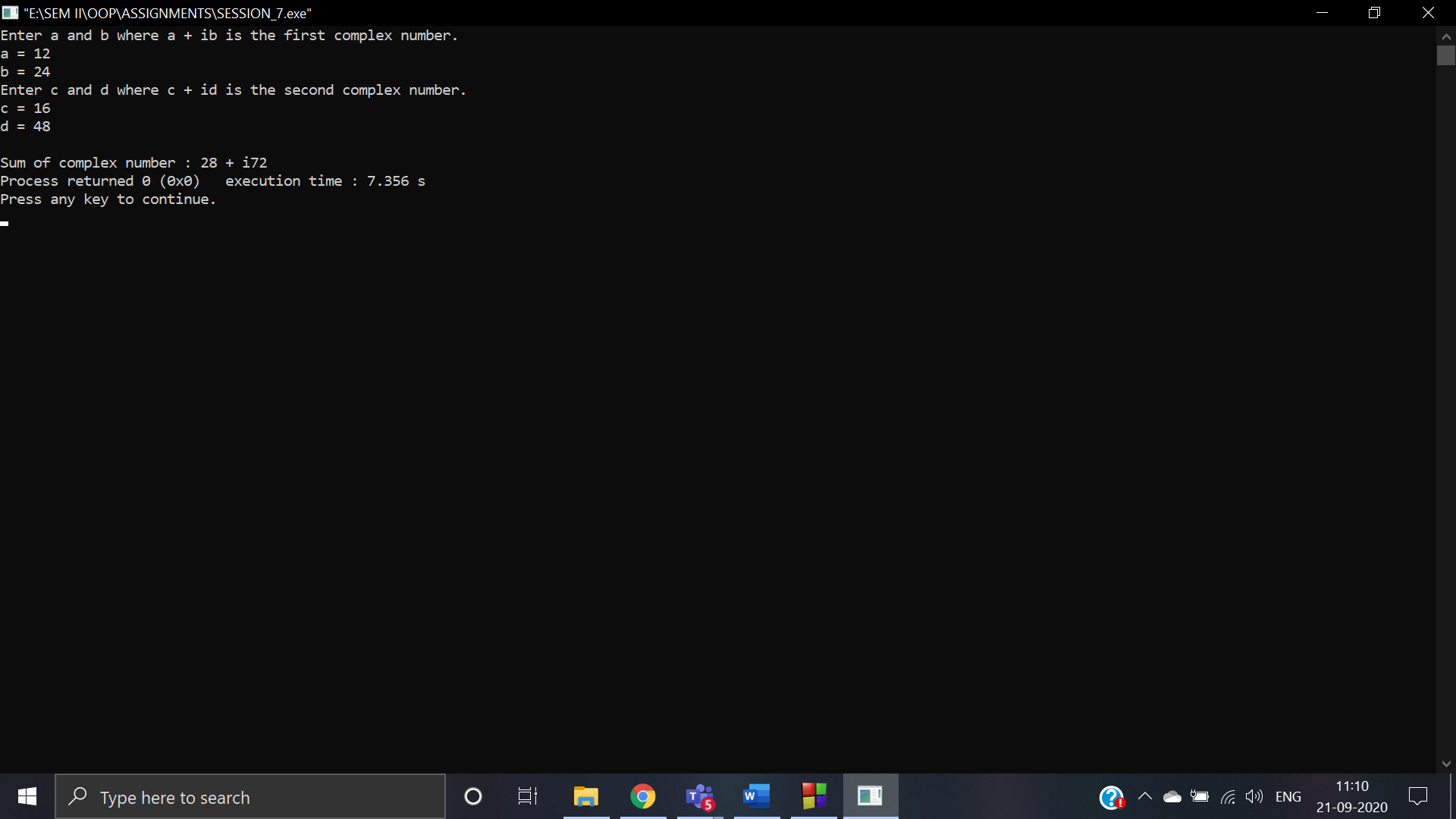
c = c.addComp(a, b);

cout<<"\nSum of complex number : "<< c.real << " + i"<<c.img;

return 0;

}

**OUTPUT:**



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

**ANS:**

**CODE:** #include<iostream>

#include<math.h>

using namespace std;

int main()

{

int x;

cout<<"Enter the number whose square root you want::";

cin>>x;

double y = sqrt(x);

try

{

if(x<0)

{

throw x;

}

else

{

cout<<"\nThe square root of "<<x<<"is::"<<y;

}

}

catch(int ex)

{

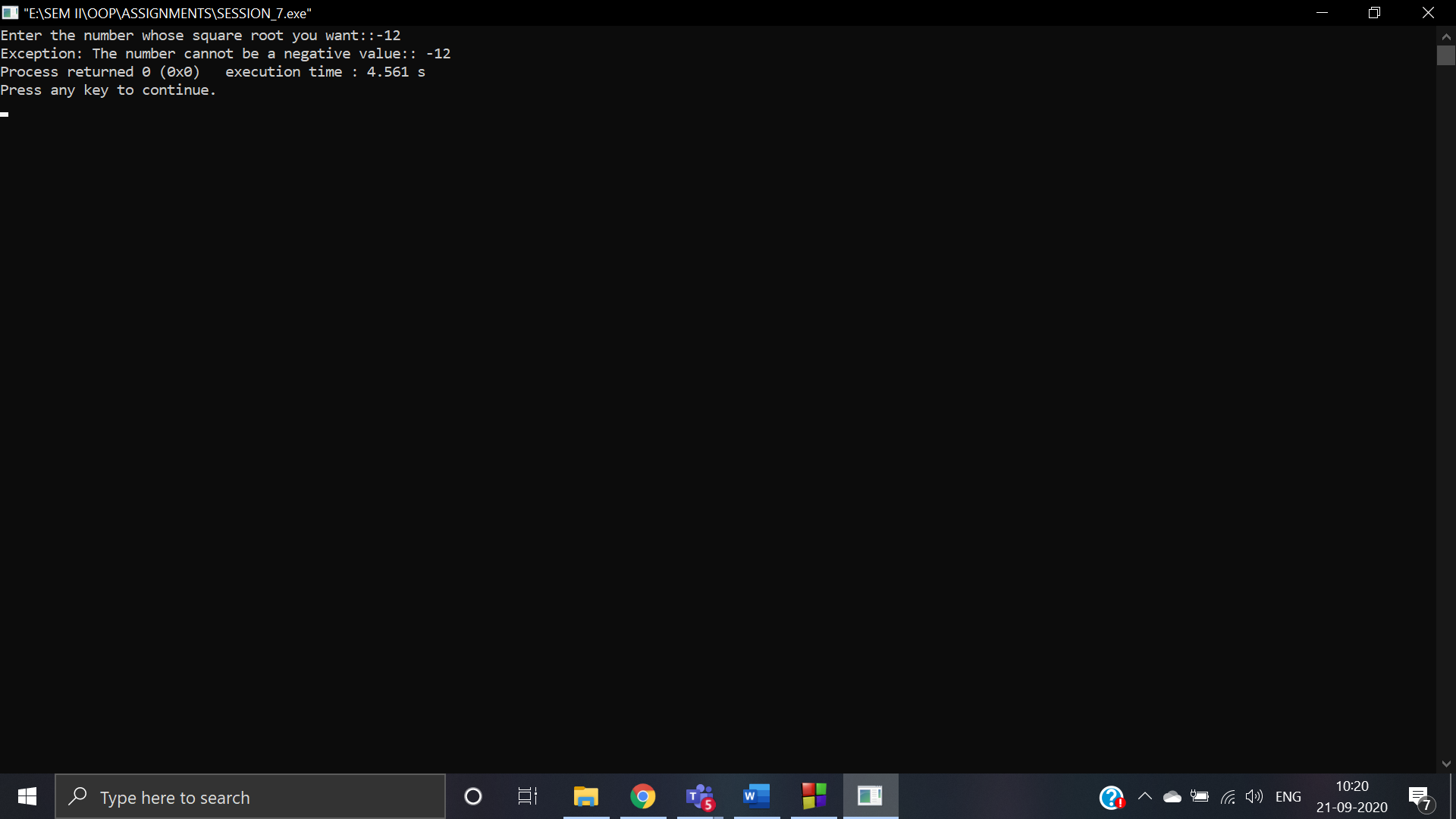
cout<<"Exception: The number cannot be a negative value:: "<<ex;

}

return 0;

}

**OUTPUT:**



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

**ANS:**

**CODE:** #include<iostream>

using namespace std;

int main()

{

int age;

cout<<"\tVOTING BARRIER CHECK";

cout<<"\n-------------------------------------";

cout<<"\nEnter your age::";

cin>>age;

try

{

try

{

if(age<0)

{

throw(age);

}

}

catch(int a)

{

cout<<"\nException: The entered age is negative.... Plz enter a non-negative number::"<<a;

}

if(age<18||age>125)

{

throw(age);

}

else

{

cout<<"\nYou are eligible to vote ;)";

}

}

catch(int ag)

{

cout<<"\nException: You are not eligible to vote due to your age is below the voting age::"<<ag;

}

return 0;

}

**OUTPUT:**

