Institute of Computer Technology
B. Tech Computer Science and Engineering
Subject: ESFP-II (2CSE203)

PRACTICAL-13

AIM: - To learn about exception handling in C++.

1. Mr. John want to write an exception handling class program in C++, for that accept age of a man from user and check age is greater than or equal to 18 or not, if yes then that man is eligible for vote otherwise throw exception. [Implement the concept of try, catch and throw concept.]

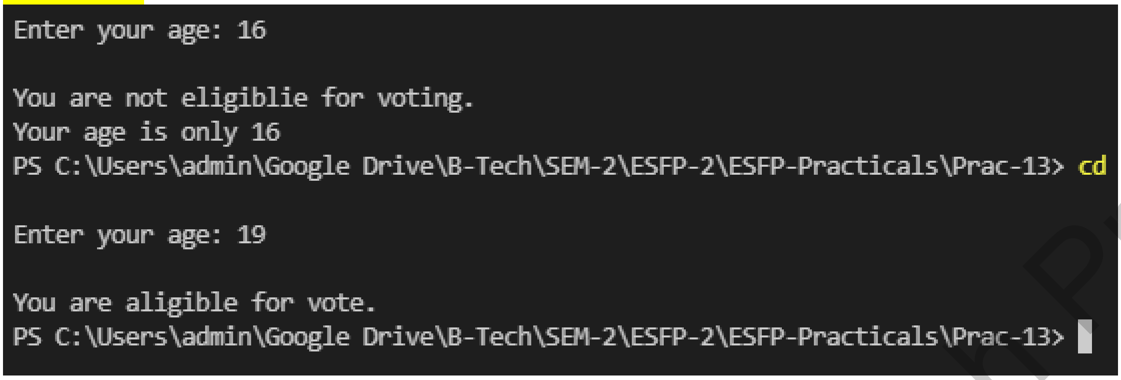
Input: Enter your age :45
Output: You are eligible for vote.
Input: Enter your age: 17
Output: Exception: You are not eligible for vote: Your age is only 17.

CODE:

```
#include <iostream>
#include <exception>
#include<stdexcept>

using namespace std;

int main()
{
   int age;
   cout<<"\nEnter your age: ";
   cin>>age;
   try
   {
      if (age>=18)
      {
         cout<<"\nYou are aligible for vote.";
      }

      else if(age < 18)
      {
         throw age;
      }
   }
   catch(int age)
   {
      cout<<"\nYou are not eligiblie for voting."<<'\n';
```

```
        cout<<"Your age is only "<<age;
    }

    return 0;
}
```

OUTPUT:



```
Enter your age: 16

You are not eligiblie for voting.
Your age is only 16
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-13> cd

Enter your age: 19

You are aligible for vote.
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-13>
```

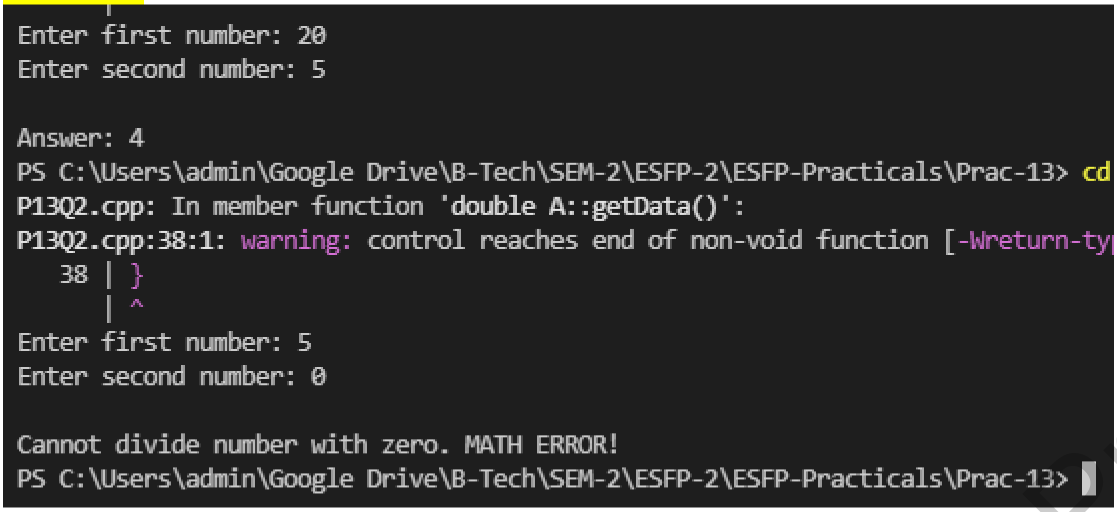
2. Write a class program with the following:

1. **A function to read two double type number from keyboard.**
2. **A function to calculate the division of the two number.**
3. **A try block to throw an exception when a wrong type of data is keyed in.**
4. **A try block throw an exception, if the condition "divide - by - zero" occurs.**
5. **Appropriate catch block to handle the exception thrown.**

CODE:

```
#include <iostream>
#include <exception>
#include <stdexcept>

using namespace std;
class A{
    public:
    double num1, num2, ans;
double getData()
{
    int j;
        cout<<"Enter first number: ";
        cin>>num1;
        cout<<"Enter second number: ";
        cin>>num2;
    try
    {
        if (cin.fail())
        {
            throw j;
        }
        if(num2==0)
        {
            throw num2;
        }
    }
    catch(double num2)
    {
        cout<<"\nCannot divide number with zero. MATH ERROR!";
        return 0;
    }
    catch(int j)
    {
        cout<<"\nIncompatible datatype entered.";
        return 0;
    }
    divide();
}

double divide()
{
    ans=num1/num2;
    cout<<"\nAnswer: "<<ans;
    return 0;
}
};
int main()
{
    A obj;
    try
    {
        obj.getData();
    }
    catch(...)
    {
        cout<<"\nEXCEPTION HANDLED";
    }
    return 0;
}
```

OUTPUT:



```
Enter first number: 20
Enter second number: 5

Answer: 4
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-13> cd
P13Q2.cpp: In member function 'double A::getData()':
P13Q2.cpp:38:1: warning: control reaches end of non-void function [-Wreturn-ty
   38 | }
      | ^
Enter first number: 5
Enter second number: 0

Cannot divide number with zero. MATH ERROR!
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-13>
```

Post Practical Task

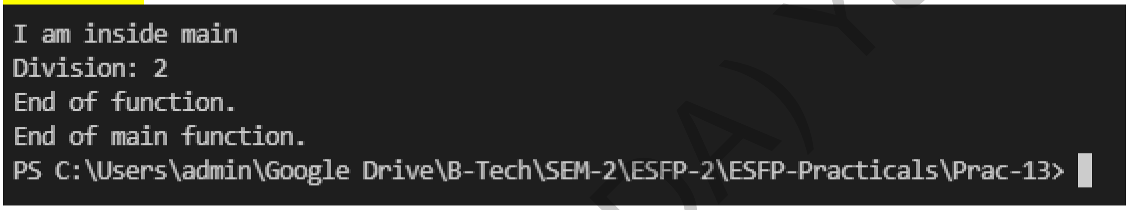
1. Make a program, to demonstrate the concept of rethrowing mechanism in an exception handling.

CODE:

```
#include <iostream>
#include <exception>
#include<stdexcept>

using namespace std;
void abc(int a,int b)
{
   try
   {
      if(b==0)
      {
         throw b;
      }
      else
      {
         cout<<"\nDivision: "<<(a/b);
      }
   }
   catch(int i)
   {
      cout<<"\nCaught interger inside function "<<i;
      throw;
   }
   cout<<"\nEnd of function.";
}
```

```
int main()
{
   cout<<"\nI am inside main";
   try
   {
      {
         abc(10,5);
      }
   }
   catch(int num)
   {
      cout<<"\nCaught integer inside main "<<num;
   }
   cout<<"\nEnd of main function.";
   return 0;
}
```

OUTPUT:



```
I am inside main
Division: 2
End of function.
End of main function.
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-13>
```

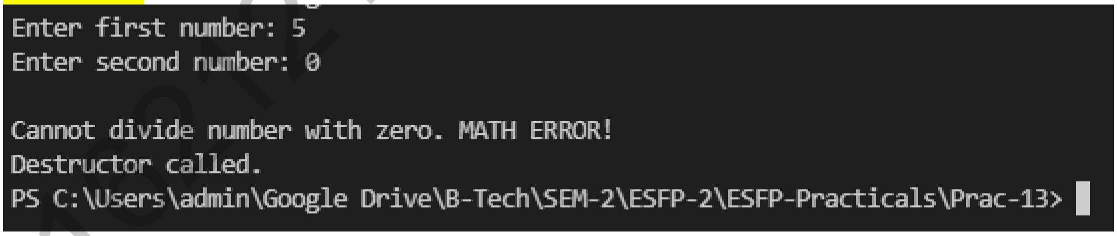
2. Make a program using exception handling in C++, where you have to show the use Implementation of exception using class constructor and destructor.

CODE:

```
#include <iostream>
#include <exception>
#include<stdexcept>

using namespace std;

class A{
   int num1,num2;
   public:
   A()
   {
      int j;
      cout<<"Enter first number: ";
      cin>>num1;
      cout<<"Enter second number: ";
      cin>>num2;
```

```
        try
        {
           if(num2==0)
           {
              throw num2;
           }
           else
           {
              cout<<"\nAnswer: "<<num1/num2;
           }
        }
        catch(int num2)
        {
           cout<<"\nCannot divide number with zero. MATH ERROR!";
        }
     }

     ~A()
     {
        cout<<"\nDestructor called."<<endl;
     }
  };

  int main()
  {
     A a;
     return 0;
  }
```

OUTPUT:



```
Enter first number: 5
Enter second number: 0

Cannot divide number with zero. MATH ERROR!
Destructor called.
PS C:\Users\admin\Google Drive\B-Tech\SEM-2\ESFP-2\ESFP-Practicals\Prac-13>
```

3. Find output from given below program:

```
#include <iostream>
#include <exception>
using namespace std;
class Test: public exception
{
```

```
virtual const char* what() const throw()
{
return "Exception arised";
}
} obj;
int main ()
{
try
{
throw obj;
}
catch (exception& e)
{
cout << e.what() << endl;
}
return 0;
}
```

a. Exception arised

b. Compile time error

c. Run time error

d. None of the above.

5. Find output from the given below program:

```
# include <iostream>
using namespace std;
double div(int a, int b)
{
if ( b == 0 )
{
throw "Denominator never be zero for division!";
}
return (a / b);
}
int main ()
{
int x = 10; int y = 0;
double z = 0;
try
{
z = div(x, y);
cout << z << endl;
}
catch (const char* ch)
{
cout << ch << endl;
}
return 0;
}
```

a. 2

b. 50

c. Denominator never be zero for division!

d. None of the above.