Institute of Computer Technology

B. Tech. Computer Science and Engineering

Sub: ESFP – II

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.

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.

Post Practical Task

1. Make a program, to demonstrate the concept of rethrowing mechanism in an exception handling.
2. Make a program using exception handling in C++, where you have to show the use

   Implementation of exception using class constructor and destructor.
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.