Page No.
Date

Assignment No : 6

1] syntax of try and catch with appropriate example.

→

Syntax of try & catch :

```
try {
    // Block of code to try
    throw exception;
}
catch () {
    // Block of code to handle errors
}
```

e.g :

```
try {
    int age = 20;
    if (age >= 18) {
        cout << " eligible for driving ";
    }
    else {
        throw (age);
    }
}
catch (int no) {
    cout << "not eligible for driving at least
        your age should be 18. ";
    cout << " age is : " << no;
}
```

Page No.
Date

2] Advantages of using exception handaling.

→ Advantages:

(1) exception handaling can control run time errors that occur in the program.

(2) It can avoid abnormal termination of the program also shows the behaviour of the program to users.

(3) It can seperate the error handling code and normal code by using try-catch block.

(4) It developes a powerful coding which ensures that exceptions can be prevented.

3] Exception specification

→ Exception specification are a c++ language feature that indicate the programmer's intent about the exception types that can be propogated by a function. You can specify that a function may or may not exit by an exception by using an exception specification. The compilers can use this information to optimize calls to the function, and to terminate the program if an unexpected exception escapes the function.

4] Multiple c++ exception classes

→ In c++ standard exceptions are defined in <exception> class that we can use inside our programs.

All the exception classes in c++ are derived from std :: exception class.

Page No.
Date

following are c++ common exception classes:

exception	Description.
std :: exception	→ It is an exception and parent class of all standard c++ exceptions.
std :: logic_failure	→ It is an exception that can be detected by reading a code.
std :: runtime_error	→ It is an exception that cannot be detected by reading a code.
std :: bad_cast	→ This exception is generally be thrown by dynamic_cast
std :: bad_typeid	→ This exception is generally be thrown by typeid.
std :: bad_alloc	→ This exception is generally be thrown by new.

5] conclusion :

→ understood the concept of exception handling and how to implment the concept of exception handling.

CODE:

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
    int i = 0, n, j;
xyz:
    cout
        << "note: the size should be a positive interger and less than 3." <<
endl
        << "enter the size of matrix:";
    cin >> n;
    try
    {
        if (n >= 4 || n <= 0)
        {
            throw n;
        }
    }
    catch (int p)
    {
        if (p >= 4)
        {
            cout << "\n\t\t !!!exception occurred!!!!" << endl
                 << "\tthe size should not be greater than 3. " << endl
                 << endl;
            goto xyz;
        }
        else
        {
            cout << "\n\t\t !!!exception occurred!!!!" << endl
                 << "\tthe size should not be zero or less than zero." << endl
                 << endl;
            goto xyz;
        }
    }
    int arr1[n][n], arr2[n][n];
    cout << "enter the elements of matrix 1:" << endl;
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            cin >> arr1[i][j];
        }
    }
    cout << "enter the elements of matrix 2:" << endl;
    for (i = 0; i < n; i++)
```

```
    {
        for (j = 0; j < n; j++)
        {
            cin >> arr2[i][j];
        }
    }

    int choice, sum[n][n], sub[n][n], mult[n][n];
    while (1)
    {
        cout << "enter 1)addition 2)subtraction 3)multiplicatin " << endl;
        cin >> choice;
        switch (choice)
        {
        case 1:
            cout << "the addition of matrix 1 and matrix 2 is:" << endl;
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < n; j++)
                {
                    sum[i][j] = arr1[i][j] + arr2[i][j];
                    cout << sum[i][j] << " ";
                }
                cout << endl;
            }
            break;
        case 2:
            cout << "the subtraction of matrix 1 and matrix 2 is:" << endl;
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < n; j++)
                {
                    sub[i][j] = arr1[i][j] - arr2[i][j];
                    cout << sub[i][j] << " ";
                }
                cout << endl;
            }
            break;
        case 3:
            cout << "the multiplication of matrix 1 and matrix 2 is:" << endl;
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < n; j++)
                {
                    mult[i][j] = 0;
                    for (int k = 0; k < n; k++)
                    {
                        mult[i][j] = mult[i][j] + (arr1[i][k] * arr2[k][j]);
```

```
                    }
                    cout << mult[i][j] << " ";
                }
                cout << endl;
            }
            break;
        case 4:
            return 0;
            break;
        default:
            cout << "wrong choice" << endl;
            break;
        }
    }
    return 0;
}
```

OUTPUT:

```
note: the size should be a positive interger and less than 3.
enter the size of matrix:4

                !!!exception occurred!!!!
        the size should not be greater than 3.

note: the size should be a positive interger and less than 3.
enter the size of matrix:2
enter the elements of matrix 1:
1
2
4
3
enter the elements of matrix 2:
2
4
1
2
enter 1)addition 2)subtraction 3)multiplicatin
3
the multiplication of matrix 1 and matrix 2 is:
4 8
11 22
enter 1)addition 2)subtraction 3)multiplicatin
1
the addition of matrix 1 and matrix 2 is:
3 6
5 5
enter 1)addition 2)subtraction 3)multiplicatin
0
wrong choice
enter 1)addition 2)subtraction 3)multiplicatin
4
PS D:\program\secondyear>
```