Exception Handling

1. Write a C++ program to demonstrate the use of try, catch block with the argument as an integer and string using multiple catch blocks.

```
#include <iostream>
using namespace std;

void test(float);

int main()
{
    test(-5);
    return 0;
}

void test(float x)
{
    try
    {
        if (x < 0)
            throw x;
        else
            cout << "It is a positive number";
        throw;
    }
    catch (float a)
    {
        cout << "It is a non-positive number" << endl;
    }
    catch (char a[100])
    {
        cout << a << endl;
    }
}
===================================================================
Output:
It is a non-positive number
```

2. Write a C++ program to demonstrate try, throw and catch statements.

```
#include <iostream>
using namespace std;

int main()
{
    try
    {
        throw 'e';
    }
    catch (char a)
    {
        cout << "Exception catch '" << a << "'" << endl;
    }
    return 0;
}
```

```
==============================================================================
Output:
Exception catch 'e'
```

3. Write a C++ program to perform arithmetic operations on two numbers and throw an exception if the dividend is zero or does not contain an operator.

```
#include <iostream>
using namespace std;

int main()
{
    float a, b; // for numbers
    char ch;    // for operator
    try
    {
        cout << "Enter a number : ";
        cin >> a;
        if (a == 0)
            throw 0; // Exceptio throw integer value
        cout << "Enter operator to perform operation (+, -, *, /): ";
        cin >> ch;
        if (ch != '+' && ch != '-' && ch != '/' && ch != '*')
            throw ch; //  Exceptio throw character value
        cout << "Enter second number: ";
        cin >> b;

        switch (ch)
        {
        case '+':
            cout << "Sum is " << (a + b) << endl;
            break;
        case '-':
            cout << "Subtraction is " << (a - b) << endl;
            break;
        case '/':
            if (b == 0)
                throw 0; // Exceptio throw integer value
            cout << "Division is " << (a / b) << endl;
            break;
        case '*':
            cout << "Multiplication is " << (a * b) << endl;
            break;
        default:
            cout << "Invalid Input" << endl;
        }
    }
    catch (const int integer)
    {
        cout << integer << " is not allow" << endl;
    }
    catch (const char ch)
    {
        cout << ch << " is not allow for operator" << endl;
    }

    return 0;
}
==============================================================================
Output:
Enter a number : 10
Enter operator to perform operation (+, -, *, /): +
Enter second number: 0
Sum is 10
```

4. Write a C++ program to accept an email address and throw an exception if it does not contain @ symbol.

```
#include <iostream>
using namespace std;
int isEmailValid(char *a);

int main()
{
    char email[100];
    cout << "Enter email address: ";
    cin.getline(email, 100);
    try
    {
        if (isEmailValid(email))
            cout << "Email is valid" << endl;
        else
            throw 0;
    }
    catch (const int x)
    {
        cout << "Invalid Email-id " << endl;
    }
    catch (...)
    {
        cout << "\nDefault Exception";
    }

    return 0;
}

int isEmailValid(char *a)
{
    int AtOffset = -1;
    int DotOffset = -1;
    int length = 0;
    for (int i = 0; a[i] != '\0'; i++)
    {
        if (a[i] == '@')
            AtOffset = i;
        else if (a[i] == '.')
            DotOffset = i;
        length++;
    }
    if (AtOffset == -1 || DotOffset == -1)
        return 0;
    if (AtOffset > DotOffset)
        return 0;
    return !(DotOffset >= (length - 1));
}
=============================================================================
Output:
Enter email address: akhtar.frs@dsd
Invalid Email-id
```

5. Write a C++ program to accept a mobile number and throw an exception if it does not contain 10 digits.

```
#include <iostream>
#include <string.h>
using namespace std;

int main()
```

```
{
    string Mobile;
    int digit = 0;
    cout << "Enter mobile number: ";
    cin >> Mobile;
    cout << Mobile << endl;
    try
    {
        for (int i = 0; Mobile[i]; i++)
            ++digit;
        if (digit == 10)
            throw 1;
        else
            cout << "Invalid mobile number";
        throw;
    }
    catch (int x)
    {
        cout << "Valid Mobile Number" << endl;
    }
    catch (char a[20])
    {
        cout << a << endl;
    }
    return 0;
}
==============================================================================
Output:
Enter mobile number: 7869577899
7869577899
Valid Mobile Number
```

6. Write a C++ program to accept area pin code and throw an exception if it does not contain 6 digits.

```
#include <iostream>
using namespace std;

int main()
{
    string areaPin;
    int digit = 0;
    cout << "Enter area pin code: ";
    cin >> areaPin;
    cout << areaPin << endl;
    try
    {
        for (int i = 0; areaPin[i]; i++)
            ++digit;
        if (digit == 6)
            throw 1;
        else
            cout << "Invalid area pin code";
        throw;
    }
    catch (int x)
    {
        cout << "Valid area pin code" << endl;
    }
    catch (char a[20])
    {
        cout << a << endl;
    }
    return 0;
}
==============================================================================
Output:
Enter area pin code: 123546987
123546987
Invalid area pin code
```

7. Write a C++ program to accept a username if the username has less than 6 characters or does contain any digit or special symbol.

```
#include <iostream>
using namespace std;

int main()
{
    string userName;
    int char_count = 0;
    int check_digit = 0;
    int special_symbol = 0;
    cout << "Enter username: ";
    cin >> userName;
    cout << userName << endl;
    try
    {
        for (int i = 0; userName[i]; i++)
        {
            ++char_count;
            if (i < 6)
            {
                if(userName[i] >= 48 && userName[i] <= 57 )
                {
                    check_digit = 1;
                }
                if(userName[i] == '@' || userName[i] == '-' || userName[i] ==
'_' )
                {
                    special_symbol = 1;
                }
            }
        }

        if (char_count < 6 && check_digit == 1 && special_symbol == 1)
            throw 1;
        else
            cout << "Username not accepted";
        throw;
    }
    catch (int x)
    {
        cout << "Username Accepted" << endl;
    }
    catch (char a[20])
    {
        cout << a << endl;
    }
    return 0;
}
==============================================================================
Output:
Enter username: she@8
she@8
Username Accepted
```

8. Write a C++ program to accept a password and throw an exception if the password has less than 6 characters or does not contain a digit or does not contain any special character or does not contain any capital letter.

```
#include <iostream>
using namespace std;

int main()
{
    string userName;
    int char_count = 0;
    int check_digit = 0;
    int special_symbol = 0;
    int check_capital = 0;
    cout << "Enter password: ";
    cin >> userName;
    cout << userName << endl;
    try
    {
        for (int i = 0; userName[i]; i++)
        {
            ++char_count;
            if (userName[i] >= 48 && userName[i] <= 57)
                check_digit = 1;
              if (userName[i] == '@' || userName[i] == '-' || userName[i] ==
'_')
                special_symbol = 1;
            if (userName[i] >= 'A' && userName[i] <= 'Z')
                check_capital = 1;
        }

          if (char_count >= 6 && check_digit == 1 && special_symbol == 1 &&
check_capital == 1)
            throw 1;
        else
            cout << "Password not accepted";
        throw;
    }
    catch (int x)
    {
        cout << "Password Accepted" << endl;
    }
    catch (char a[20])
    {
        cout << a << endl;
    }
    return 0;
}

=============================================================================
Enter password: Mukesh@007
Mukesh@007
Password Accepted
```

9. Write a C++ program to accept Gmail id only and throw an exception if the id does not contain @ and gmail.com.

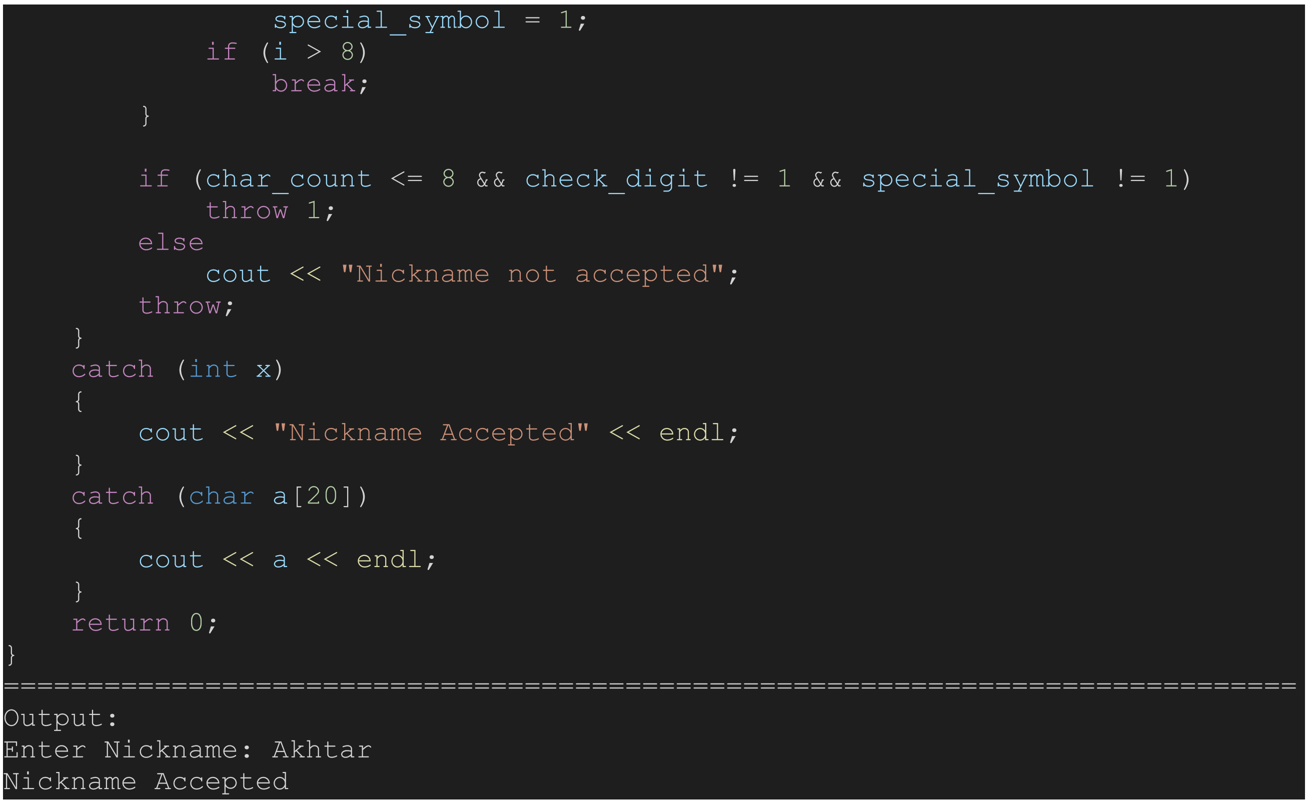
```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string email;
    string gmail = "@gmail.com";
    cout << "Enter gmail: ";
    cin.ignore();
    cin >> email;
    try
    {
        if (email.find(gmail) != -1)
        {
            throw 1;
        }
        else
        {
            cout << "Gamil not Accepted" << endl;
            throw;
        }
    }
    catch (int x)
    {
        cout << "Gmail Accepted" << endl;
    }
    catch (char *p)
    {
        cout << p;
    }
    catch (...)
    {
        cout << "Other Exception" << endl;
    }
    return 0;
}
===================================================================
Output:
Enter gmail: mgajendra@.com
Gamil not Accepted
```

10. Write a C++ program to accept Nickname and throw an exception if it has greater than 8 characters or does contain a digit or special symbol or space.

```
#include <iostream>
using namespace std;

int main()
{
    char Nickname[100];
    int char_count = 0;
    int check_digit = 0;
    int special_symbol = 0;
    cout << "Enter Nickname: ";
    cin.getline(Nickname, 100);
    try
    {
        for (int i = 0; Nickname[i]; i++)
        {
            ++char_count;
            if (Nickname[i] >= 48 && Nickname[i] <= 57)
                check_digit = 1;
              if (Nickname[i] == '@' || Nickname[i] == '-' || Nickname[i] ==
'_' || Nickname[i] == ' ')
```



```
                special_symbol = 1;
            if (i > 8)
                break;
        }

        if (char_count <= 8 && check_digit != 1 && special_symbol != 1)
            throw 1;
        else
            cout << "Nickname not accepted";
        throw;
    }
    catch (int x)
    {
        cout << "Nickname Accepted" << endl;
    }
    catch (char a[20])
    {
        cout << a << endl;
    }
    return 0;
}
==============================================================================
Output:
Enter Nickname: Akhtar
Nickname Accepted
```