EXCEPTION HANDLING

Exceptions are errors that occur at run time.
Exceptions are run time anomalies or unusual conditions that a program may encounter while executing.

Anomalies (Unexpected situations) might include:

1. Division by zero.
2. An array outside of its bound.
3. Running out of memory.

The error handling code that performs the following tasks:

a. Find the problem (with the exception)
b. Inform that an error has occurred (throw the exception)
c. Receive the error information (Catch the exception)
d. Take corrective actions (Handle the exception)

Error Handling Code consists of two segments:

1. To detect error and to throw exception.
2. To catch the exception and to take appropriate actions

<u>C++ Exception Handling Mechanism is built upon three keywords</u>:

Try
Throw
Catch

1. **Try block:** This block contains program statements which may <u>generate exception.</u>

2. **Throw block:** When an <u>exception is detected</u> it is thrown using throw statement in try block.

3. **Catch block:** Catches the exception thrown by the throw statement in the try block and handle it appropriately.
   Syntax:

```
try
{

   Throw exception;     //block of statements which detects and
                          Throws an exception
}
catch (type arg)       // Catches the exception
{

   Block of statements;    // that handles the exception

}
```

For Multiple catches:

```
try {
// try block
}
catch (type1 arg) {
// catch block
}
catch (type2 arg) {
// catch block
}
catch (type3 arg) {
// catch block
}
..
.
catch (typeN arg) {
// catch block
}
```

Catching all exceptions:

```
Catch (…)
{
// catch block
}
```

Program on try block throwing an exception

```
   main()
   {
   int a,b;
   cout<<” enter a and b values\n”
   cin>>a>>b;

   int x;
 x=a-b;
    try
   {
   if (x!=0)
   {
   cout<<”result a/x”<<a/x<<endl;
   }
   else
   {
   throw x;
   }
   }
   catch( int i)
   {
   cout<<”exception caught\n”;
   }
   cout<<”end”;
   }
```

Output:

First run:

```
Enter a and b values
20      15
```

```
Result a/x   4
End

Second run:
Enter a and b values
15     15
Exception caught
End
```

```
// multiple catch block (throwing different type of exceptions)

Void test(int );
main()
{
Test(0);
Test(1);
Test(-1);
}
Void test(int x)
{
Try
{
If(x==0)
Throw 1; // integer type
If(x==1)
Throw ‘x’; // character type
If(x==-1)
Throw 1.0;  // float type
}
}
Catch(int i)
{
Cout<<”exception caught\n”;
}
Catch(char j)
{
Cout<<”exception caught\n”;
}
Catch(float k)
{
Cout<<”exception caught\n”;
}
```

```
Exception caught
Exception caught
Exception caught
```