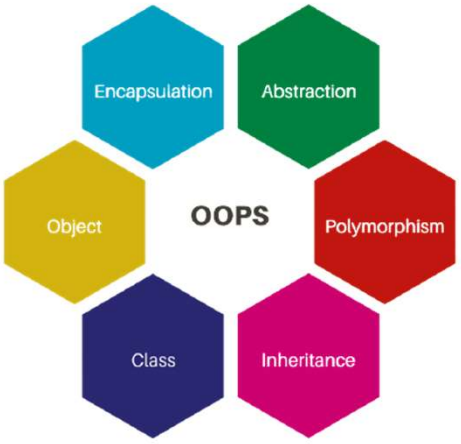


ABESIT
COLLEGE OF ENGINEERING
CRC-Training

Java™

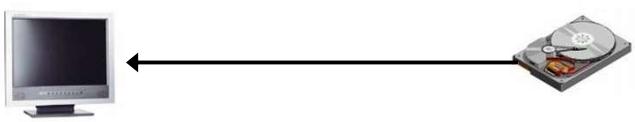
Exception Handling

ABESIT
COLLEGE OF ENGINEERING
CRC-Training

Why use Exception?

- In earlier programming language, the things that leads to confusion
  - Error detection
  - Error reporting
  - Error Handling

Take an Example of reading an entire file into memory



ABESIT COLLEGE OF ENGINEERING
CRC-Training

Why use Exception?

```
readFile
{
open the file;
determine its size;
allocate that much memory;
read the file into memory;
close the file;
}
```

Steps To Be Followed For Reading a File in the memory

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Why use Exception?

Can U Guess the potential Errors in this operation

- What happens if the file can't be opened?
- What happens if the length of the file can't be determined?
- What happens if enough memory can't be allocated?
- What happens if the read fails?
- What happens if the file can't be closed?

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Why use Exception?

```
errorCodeType
readFile
{
initialize errorCode = 0;
open the file;
 if (theFileIsOpen)
{
 determine the length of the file;
 if (gotTheFileLength)
 {
 allocate that much memory;
 if (gotEnoughMemory)
 {
```

```
read the file into memory;
 if (readFailed)
 {
 errorCode = -1;
 }
 }
 else
 {
  errorCode = -2;
 }
      }
 else
  {
  errorCode = -3;
 }
```

```
close the file;
  if (theFileDidntClose && errorCode == 0)
 {
  errorCode = -4;
 }
  else
  {
  errorCode = errorCode and -4;
 }
 }
else
{
  errorCode = -5;
}
   return errorCode;
}
```

To Answer this question your program should have error detection, reporting, and handling mechanism that will look like:

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Why use Exception?

```
readFile {
  try {
open the file;
determine its size;
allocate that much memory;
read the file into memory;
close the file;
}catch (fileOpenFailed)
{
  doSomething;
}
catch(sizeDeterminationFailed)
 {
 doSomething;
 }
```

Exception helps To Solve this problem

```
catch (memoryAllocationFailed)
 {
doSomething;
}
catch (readFailed)
{
doSomething;
}
catch (fileCloseFailed)
{
      doSomething;
}
}
```

ABESIT COLLEGE OF ENGINEERING
CRC-Training

What is an Exception?

- An exception can be defined as an abnormal event that occurs during program execution and disrupts the normal flow of instructions.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Uses of Exception?

- Consistent error reporting style.
- Easy to pass errors up the stack.
- Pinpoint errors better in stack trace.
  - As long as you “fail fast” and throw as soon as you find a problem.
- Wrap useful information up in your exception classes, use that information later when handling the exception.
- Exceptions don’t always have to be errors, maybe your program can cope with the exception and keep going!
- Separating error handling code from Regular code

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Types of Errors

1. Compile Time Error

Compile Time Error – Occurs during compilation of a program which includes syntax errors, semantic errors etc.

2. Run Time Error

Runtime Error – Generated while running the program. For example, program that connects to another machine to fetch some data but found the target machine off, a error will encounter.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Exception Occurrence Reason

1. Running out of memory.
2. Resource allocation errors.
3. Inability to find files.
4. Problem in network connectivity.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Exception Occurrence Levels

- Hardware/operating system level.
  - ✓ Arithmetic exceptions; divide by 0, under/overflow.
  - ✓ Memory access violations; segment fault, stack over/underflow.
- Language level.
  - ✓ Type conversion; illegal values, improper casts.
  - ✓ Bounds violations; illegal array indices.
  - ✓ Bad references; null pointers.
- Program level.
  - ✓ User defined exceptions.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Categories of Exceptions

1. Built in Exceptions
   - 1.a Checked Exceptions
   - 1.b Unchecked Exceptions
2. User Defined Exceptions

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Checked Exceptions

- Checked exceptions are so called because both the Java compiler and the Java virtual machine check to make sure this rule is obeyed.
- The checked exceptions that a method may raise are part of the method's signature.
- Every method that throws a checked exception must advertise it in the throws clause in its method definition, and
- Every method that calls a method that advertises a checked exception must either handle that exception (with try and catch) or must in turn advertise that exception in its own throws clause.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

UnChecked Exceptions

- Raised implicitly by system because of illegal execution of program.
- Do not need to announce the possibility of exception occurrence.
- When unchecked exception occurred, if programmer does not deal with it, it would be processed by default exception handler.
- Exception extended from Error class and RuntimeException class.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Implementing Exception Handling

1. try
2. catch
3. throws
4. throw
5. finally

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Implementing Exception Handling

- Using try and catch statements
  - ✓ The try block encloses the statements that might raise an exception within it and defines the scope of the exception handlers associated with it.
  - ✓ The catch block is used as an exception-handler. You enclose the code that you want to monitor inside a try block to handle a run time error.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Implementing Exception Handling using try & catch

```
try
{
        // Statements that cause an exception.
}
catch(ExceptionName obj)
{
        // Error handling code.
}
```

Example

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Implementing Exception Handling using try & catch

```
public class UnitRate
{
   public void calculatePerUnitRate()
   {
      int qty=20, rate=0,punit=0;
     try
      {
            punit=qty/rate;
      } catch(ArithmeticException ae)
      { System.out.println("The product rate cannot be Zero, So Per Unit Rate Displayed Below is Invalid ");
      }
       System.out.println("The Per Unit Rate is = "+punit);
   }
}
```

Arithmetic error occurred and passed to exception handler i.e.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Using Multiple catch statements / blocks

- A single try block can have many catch blocks.
- The multiple catch blocks generate unreachable code error.
- If the first catch block contains the Exception class object then the subsequent catch blocks are never executed. If this happen it is known as unreachable code problem.
- To avoid unreachable code error, the last catch block in multiple catch blocks must contain the Exception class object.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Using Multiple catch statements / blocks Example

```
try {
   // dangerous code here!
}
catch(ArithmeticException e) {
   // Specific error handling here
}
catch(RuntimeException e) {
   // More general error handling here
}
```

Order Matters!
Less general
first!

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Using finally clause

- The finally block is used to process certain statements, no matter whether an exception is raised or not.

Syntax of finally block

```
try
{
        // Block of code
}
finally
{
// Block of code that is always executed
irrespective of an exception being raised or
not.
}
```


ABESIT COLLEGE OF ENGINEERING
CRC-Training

Using finally clause

```
try {
        // …
} catch (ExceptionType1  identifier) {
        // …
} catch (ExceptionType2 identifier) {
        // …
} finally  {
        // …
}
```

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Exception Propagation

```
1.  public  class  UnitRate {
2.          void  calculatePerUnitRate() {
3.                int  qty = 25,  rate = 0,punit=0;
4.                punit = qty / rate;
5.          }
6.          void  callPerUnitRate() {
7.                calculatePerUnitRate();
8.          }
9.          public static void main(String[] args) {
10.                 UnitRate ur = new UnitRate();
11.                 ur.callPerUnitRate();
12.           }
13.   }
```

ArithmeticException Occurred

Output by Default Exception Handler

```
java.lang.ArithmeticException:   / by zero
            at UnitRate.calculatePerUnitRate(UnitRate.java:4)
            at UnitRate.callPerUnitRate(UnitRate.java:7)
            at UnitRate.main(UnitRate.java:11)
```


ABESIT COLLEGE OF ENGINEERING
CRC-Training

Exception Propagation Exercise

```
1. public class ExerciseException {
2.   public static void main(String[] args) {
3.       try {
4.          happy1();
5.         }
6.       catch(Exception e) {
7.          e.printStackTrace();
8.          }
9.       }
10.   public static void happy1() throws Exception {
11.      happy2();
12.  }
13.  public static void happy2() throws Exception {
14.      happy3();
15.  }
16.  public static void happy3() throws Exception {
17.      throw new Exception("Be Happy Exception Prevents U!");
18.  }
19. }
```

Write down the Print Stack Trace for the ExerciseException class

ABESIT COLLEGE OF ENGINEERING
CRC-Training

User Defined Exception: Creating Your Own Exceptions

- Defining your own exceptions let you handle specific exceptions that are tailor-made for your application.
- Steps to Create a User Defined Exception
  - Create a class that extend from a right kind of class from the Exception Hierarchy.
  - Let's make a DivideByZeroException for use by our UnitRate class.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

User Defined Exception: Creating Your Own Exceptions

```
public class DivideByZeroException extends ArithmeticException
{
   public DivideByZeroException()
  {
      super("Divide by 0 error");
   }
}
```

Here we extended ArithmeticException.

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Summary

- Errors can be broadly categorized into two groups on the basis of whether the compiler is able to handle the error or not, such as compile time errors and run time errors.
- An exception is a run time error that can be defined as an abnormal event that occurs during the execution of a program and disrupts the normal flow of instructions.
- In Java, the Throwable class is the superclass of all the exception classes. It is the class at the top position in the exception class hierarchy. The Exception class and the Error class are two direct subclasses of the Throwable class.
- The built-in exceptions in Java are divided into two types on the basis of the conditions where the exception is raised:
  - ✓ Checked Exceptions or Compiler-enforced Exceptions
  - ✓ Unchecked exceptions or Runtime Exceptions

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Summary

- You can handle exception using
  - ✓ try
  - ✓ catch
  - ✓ throws
  - ✓ throw
  - ✓ finally
- You use multiple catch blocks to throw more than one type of exception.
- The finally clause is used to execute the statements that need to be executed whether or not an exception has been thrown.
- The throw statement causes termination of the normal flow of control of the Java code and stops the execution of subsequent statements after the throw statement.
- The throws clause is used by a method to specify the types of exceptions the method throws.
- You can create your own exception classes to handle the situations specific to an application

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Test Your Understanding

1. The two subclasses of the Throwable class are:
   a. Exception class and Error class
   b. Exception class and Object class
   c. Error class and RunTimeException class
   d. Exception class and RunTimeException class

2. The Throwable class is the sub class of _____ class.
   a. Exception
   b. Error
   c. Object
   d. RunTimeException

ABESIT COLLEGE OF ENGINEERING
CRC-Training

Test Your Understanding

3. Which exception occurs when you try to create an object of an abstract class or interface?
   a. ClassNotFoundException
   b. IllegalAccessException
   c. InstantiationException
   d. NoSuchMethodException

Test Your Understanding

4. Consider the statements:
   Statement A: The scope of the catch block is restricted to the statements in the preceding try block only.
   Statement B: A try block must have at least one catch block that follows it immediately.

   Which of the following options is true?

   a. Statement A is true and statement B is false
   b. Statement A is false and statement B is true
   c. Both, statements A and B, are true
   d. Both, statements A and B, are false