Throw Clause

Agenda

1 Throw Clause

2 User Defined Exception

Throw Clause

Using throw

- System-generated exceptions are thrown automatically

- At times you may want to throw the exceptions explicitly which can be done using the **throw** keyword

- The exception-handler is also in the same block

- The general form of throw is:
  - throw **ThrowableInstance**

- Here, **ThrowableInstance** must be an object of type **Throwable**, or a subclass of **Throwable**

Using throw (Contd.).

```
class ThrowDemo{
  public static void main(String args[]) {
   try {
       int age=Integer.parseInt(args[0]);
       if(age < 18)
         throw new ArithmeticException();
      else
         if(age >=60)
            throw new ArithmeticException("Employee is
  retired");
   }
  catch(ArithmeticException e) {
   System.out.println(e);
  }
  System.out.println("After Catch");
    }
}
```

Match the following :

Match the content of Column A with the most appropriate content from column B :

Column A	**Column B**
a) An exception is	a) Used to throw an exception explicitly
b) Throwable	b) Caused by Dividing an integer by zero
c) ArithmeticException is	c) An event that can disrupt the normal flow of instructions
d) Catch Block	d) This class is at the top of exception hierarchy
e) “throw” is	e) Exception Handler

Quiz

Debug the code

```
import java.io.*;
class ThrowTester {
void method() throws IOException {
throw new IOException(“Method Error");
}

public static void main(String args[]) {

ThrowTester m = new ThrowTester();
m.method();
System.out.println(“main ends...");
}
}
```

User Defined Exception

User Defined Exceptions

- Java provides extensive set of in-built exceptions
- But there may be cases where we may have to define our own exceptions which are application specific

For ex: If we have are creating an application for handling the database of eligible voters, the age should be greater than or equal to 18

In this case, we can create a user defined exception, which will be thrown in case the age entered is less than 18

User Defined Exceptions (Contd.).

- While creating user defined exceptions, the following aspects have to be taken care :
  - The user defined exception class should extend from the Exception class and its subclass
  - If we want to display meaningful information about the exception, we should override the toString() method

Example on User Defined Exceptions

```
class  MyException extends Exception  {
  public MyException() {
   System.out.println("User defined
  Exception thrown");
  }
  public String toString() {
   return "MyException Object : Age cannot
  be < 18 " ;
  }
}
```

Contd..

Example on User Defined Exceptions (Contd.).

```
class MyExceptionDemo{
  static int flag=0;
  public static void main(String args[]) {
   try {
     int age=Integer.parseInt(args[0]);
     if(age < 18)
        throw new MyException();
   }
   catch(ArrayIndexOutOfBoundsException e) {
     flag=1;
     System.out.println("Exception : "+ e);
   }
```

Contd..

Example on User Defined Exceptions (Contd.).

```
  catch(NumberFormatException e) {
     flag=1;
     System.out.println("Exception : "+ e);
  }
  catch (MyExceptionClass e) {
     flag=1;
     System.out.println("Exception : "+ e);
  }
  if(flag==0)
     System.out.println("Everything is
 fine");
 }
}
```

Fill the Code …

Complete the code to print the message “Invalid Input”

```
class InvalidInputException extends Exception {
   InvalidInputException(String s) {
      //Insert the code so that null is not printed
   }
}
class Input {
   void method() throws InvalidInputException {
      throw new InvalidInputException("Invalid Input");
   }
}
class TestInput {
   public static void main(String[] args){
      try {
         new Input().method();
      }
  catch(InvalidInputException iie) {
         System.out.println(iie.getMessage());
      }
   }
}
```

Answer

- Answer : super(s)

- *String getMessage() – used to get the customized message from the exception*.

Guess the Output ...

```
class MyException extends Exception{
    String str1;
    MyException(String str2) {
       str1=str2; }
    public String toString(){
       return ("Exception: "+str1) ;}
}
class Tester{
    public static void main(String args[]){
       try{
          throw new MyException("User Defined Exception Thrown");
          // I'm throwing user defined custom exception above
       }
       catch(MyException e){
          System.out.println("Exception Handler") ;
          System.out.println(e) ;
       }
    }
}
```

```
Exception Handler
Exception: User Defined Exception Thrown
```

Summary

In this session, you were able to :

- Learn about throw clause
- Learn about user defined exception

Assignment

Thank You