VIT-AP UNIVERSITY

Course code : **CSE2005**

Course title : **Object Oriented Programming**

User Defined Exceptions

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

- 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

User Defined Exceptions

```
class CheckAge extends Exception{
        public CheckAge(){
                System.out.println("user defined exception");
        }
        public String toString(){
                return "age is invalid";
        }
}
```

User Defined Exceptions

```
public class Main {
        public static void main(String args[]) {
                try{
                        int age=-12;
                        if(age<0)
                                throw new CheckAge();
                }
                catch(CheckAge e){
                        System.out.println(e);
                }
        }
}
```

User Defined Exceptions

```
public class Main {
        public static void main(String args[]) {
                try {
                        int age = -12;
                        if (age < 0)
                                throw new CheckAge();
                } catch (CheckAge e) {
                        System.out.println(e);
                } catch (ArithmeticException e) {
                        System.out.println(e);
                }
        }  }
```

Quiz

```
//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");
        }
}
```

Quiz

```
class TestInput {
        public static void main(String[] args) {
                try {
                        new Input().method();
                } catch (InvalidInputException iie) {
                        System.out.println(iie.getMessage());
                }
        }
}
```

VIT-AP UNIVERSITY

Course code : **CSE2005**

Course title : **Object Oriented Programming**

Finally Clause

Using finally

- When an exception occurs, the execution of the program takes a non-linear path, and could bypass certain statements
- A program establishes a connection with a database, and an exception occurs
- The program terminates, but the connection is still open
- To close the connection, finally block should be used
- The finally block is guaranteed to execute in all circumstances

Using finally

- When an exception occurs, the execution of the program takes a non-linear path, and could bypass certain statements
- A program establishes a connection with a database, and an exception occurs
- The program terminates, but the connection is still open
- To close the connection, finally block should be used
- The finally block is guaranteed to execute in all circumstances

Using finally

```
public class Main {
        public static void main(String[] args) {
                try {
                        FileInputStream fin = new FileInputStream("a.txt");
                } catch (FileNotFoundException e) {
                        System.out.println("check file location");
                } finally {
                        System.out.println("finally");
                }
        }
}
```

Using finally

```
public class Main {
        public static void main(String[] args) {
                try {
                        int x=Integer.parseInt("23.45");
                } finally {
                        System.out.println("finally");
                }
        }
}
```

Significance of printStackTrace() method

- We can use the printStackTrace() method to print the program's execution stack
- This method is used for debugging

```
public class Main {
        static void m1() throws IOException{
                m2();
        }
        static void m2() throws IOException{
                throw new IOException();
        }
```

Significance of printStackTrace() method

```
public static void main(String[] args) {
        try {
                m1();
        } catch (IOException e) {
                e.printStackTrace();
        }
    }
}
```

```
java.io.IOException
        at vit.demo.Main.m2(Main.java:11)
        at vit.demo.Main.m1(Main.java:8)
        at vit.demo.Main.main(Main.java:15)
```

Quiz

```
public class Main {
        public static void main(String[] args) {
                String s=null;
                try {
                        System.out.println(s.charAt(0));
                } catch (RuntimeException e) {
                        s="welcome";
                }
                System.out.println("exception cleared");
                finally{
                        System.out.println("end");
                } } }
```

Quiz

```
public class Main {
        static void method() {
                throw new Exception();
        }
        public static void main(String[] args) {
                String s = null;
                try {
                        method();
                }
```

Quiz

```
                catch (Throwable e) {
                        try {
                                throw new Exception();
                        } catch (Exception ex) {
                                System.out.println("exception");
                        } finally {
                                System.out.println("end");
                        }
                }
        }
}
```

Summary

We have discussed about

- User Defined Exceptions
- Finally Clause