

property	Overloading	Page No.: Overriding
(1) Method names	Must be same	Must be same.
(2) Argument types	Must be different. (at least order)	Must be same. (including order)
(3) Method Signatures.	Must be different	Must be same.
(4) Return Type.	No Restriction	Must be same but this rule is applicable until 1.4 versions only. From 1.5 version onwards. Co-variant return type also allowed
(5) private, static and final methods.	can be Overloaded.	can not be Overridden.
(6) Access Modifiers.	No Restriction	We can't reduce the scope if Access modifiers but we can increase.
(7) throws clause.	No Restriction	If child class Method throws any child checked Exception compulsory parent Class method should throw the ^{same} checked Exception or its parent otherwise we will get compile-time error but there are no restriction for ^{unchecked} checked Exception
(8) Method Resolution.	Always take care by Compiler based on references type.	Always take care by JVM based on Runtime Object.
(9) Also known as	Compile Time polymorphism or static polymorphism or early binding.	Runtime polymorphism or dynamic polymorphism or late binding.

EXCEPTION HANDLING

Page No.:

INTRODUCTION :->

• An unexpected unwanted event that disturbs the normal flow of the programme is called Exception.

→ EX :- File Not Found Exception

SQLException

NullPointerException

ClassNotFoundException etc.

→ It is highly recommendate to handle the exceptions.

→ The main objective of Exception handling is graceful termination of the programme.

→ Exception handling does not means repairing an exception. We have to provide alternative way to continue rest of the programme normally. is the concept of Exception handling.

e.g.

Our prog. requirement is to read data from remote file locating at london. At run-time, if remote file is not available, our prog. should not be terminated abnormally. We have to provide some local file to continue rest of the prog. normally. This way of defining alternative is nothing but Exception Handling.

try {

Read data from Remote File

} catch (File Not Found Exception e)

}

Use local file & continue rest of the prog normally

}