1.

//When parent class doesn’t declare an exception

import java.io.\*;

class ParentClass{

//parent class not declaring any exceptions

void method(){

System.out.println("Parent class");

}

}

class ExOver extends ParentClass{

//child class declares an unchecked exception

void method() throws ArithmeticException{

System.out.println("Child class =>> overrides parent class when Parent class doesn't declare any exception and child class declares an unchecked exception.");

}

public static void main(String args[])

{

ParentClass obj = new ExOver();

obj.method();

}

}

//When parent class doesn declares an exception

import java.io.\*;

class ParentClass{

//parent class not declaring any exceptions

void method() throws RuntimeException{

System.out.println("Parent class");

}

}

class ExOver2 extends ParentClass{

//child class declares an unchecked exception

void method() throws ArithmeticException{

System.out.println("Child class =>> overrides parent class when Parent class declares an unchecked exception and child class declares an exception which is a child class of ParentClass exception");

}

public static void main(String args[])

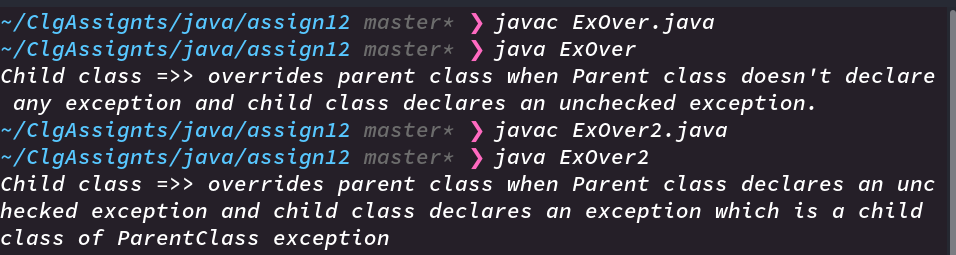
{

ParentClass obj = new ExOver2();

obj.method();

}

}



2.

import java.io.\*;

class CustomExceptionEx extends Exception

{

public CustomExceptionEx(String string)

{

super(string);

}

}

class Sample

{

public String checkNum(int num)

throws CustomExceptionEx

{

if (num == 0)

{

throw new CustomExceptionEx("Number is zero");

}

else

{

return "Number is non zero";

}

}

}

public class CustomEx

{

public static void main(String[] args)

{

Sample object = new Sample();

Sample object1 = new Sample();

try

{

//calling the method

String bar = object.checkNum(0);

}

catch (CustomExceptionEx e)

{

e.printStackTrace();

}

}

}

