What is the default behaviour if a RuntimeException occurs without using try and catch? Give an example.

Java being an object oriented programming language, whenever an error occurs while executing a statements, creates an exception object and then the normal flow of the program halts and JRE tries to find someone that can handle the raised exception. The exception object contains a lot of debugging information such as method hierarchy, line number where the exception occurred, type of exception occured , type of exception etc. When the exception occurs in a method , the process of creating the exception object and handling it over to runtime environment is called “throwing the exception”.

This small program includes an expression that intentionally causes a divide-by-zero error:

class Exc0 {

public static void main(String args[])

{

int d = 0;

int a = 42 / d;

}

}

When the Java run-time system detects the attempt to divide by zero, it constructs a new

exception object and then *throws* this exception. This causes the execution of **Exc0** to stop,

because once an exception has been thrown, it must be *caught* by an exception handler and

dealt with immediately. In this example, we haven’t supplied any exception handlers of our

own, so the exception is caught by the default handler provided by the Java run-time

system. Any exception that is not caught by your program will ultimately be processed by

the default handler. The default handler displays a string describing the exception, prints a

stack trace from the point at which the exception occurred, and terminates the program.