Exception Handling

What is an exception?

Abnormal condition that arises during runtime.

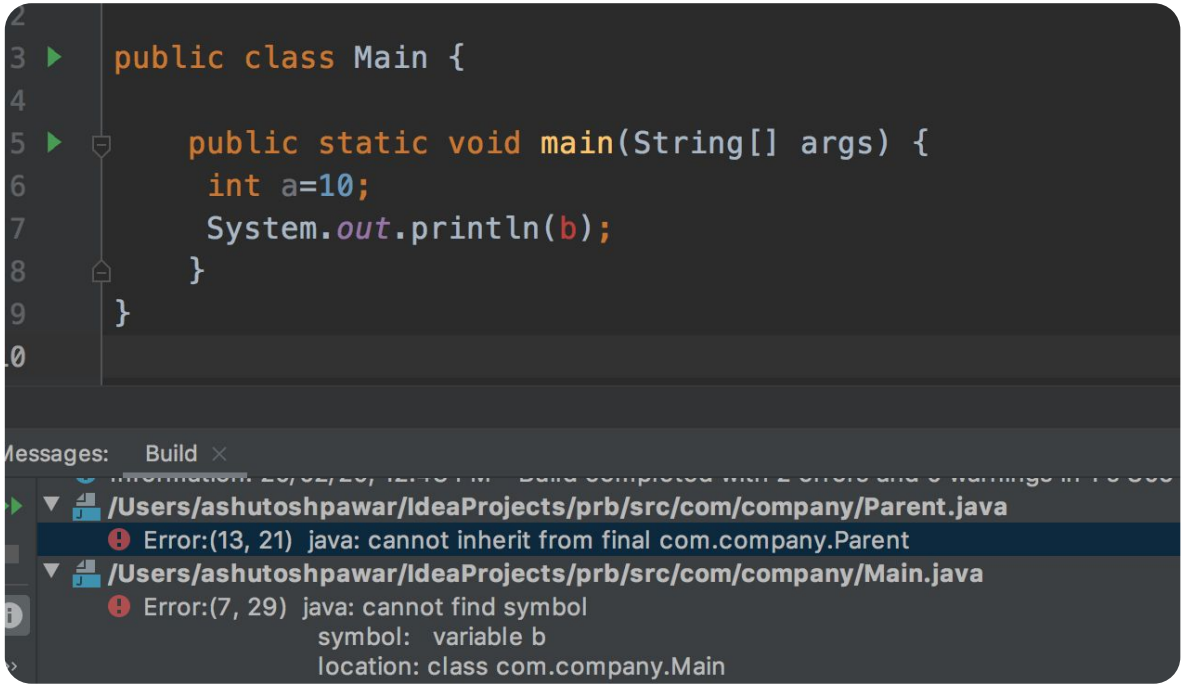
Exception is a runtime error.

Not the same as syntax error which occurs at compile-time.

Java has exception-handling mechanism.

Error vs Exception

Error


public class Main {
public static void main(String[] args) {
int a=10;
System.out.println(b);
}
}
Messages:
Build
/Users/ashutoshpawar/IdeaProjects/prb/src/com/company/Parent.java
Error:(13, 21) java: cannot inherit from final com.company.Parent
/Users/ashutoshpawar/IdeaProjects/prb/src/com/company/Main.java
Error:(7, 29) java: cannot find symbol
symbol: variable b
location: class com.company.Main

Exception


package com.company;
public class Main {
public static void main(String[] args) {
int a=10;
int b=0;
System.out.println(a/b);
}
}
Main
Run:
Main
Exception in thread "main" java.lang.ArithmeticException: / by zero
at com.company.Main.main(Main.java:8)
Process finished with exit code 1

How exceptions can be handled?

Exception is an object, that describes the exception.

When exception occurs, an object representing that expression is **thrown.**

This **thrown** exception can then be **caught** and can be handled.

Keywords used

try: A block which contains the code that might cause an exception.

catch: Catches the expression so that we can handle it.

throw: Used when we want to manually throw an exception.

finally: Any code that has to be executed after try block ends.

How exception handling looks like

```
try{
        //code that can cause an exception
}
catch{
        //code to handle exception
}
Finally{ //code that has to be executed after try block ends }
```

Let’s write some code....