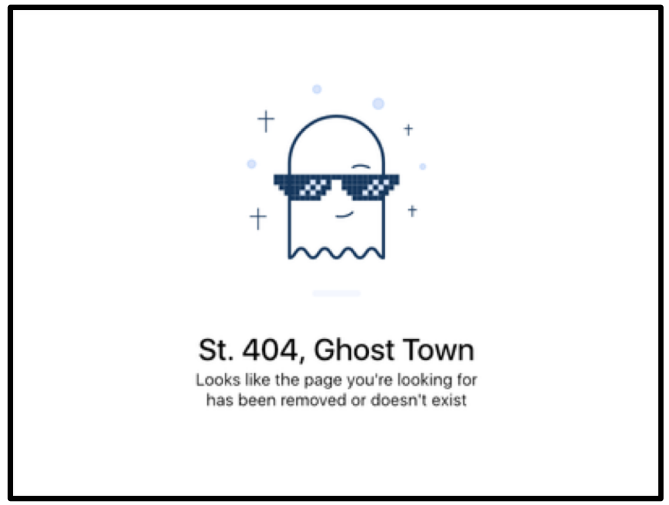
Exception Handling

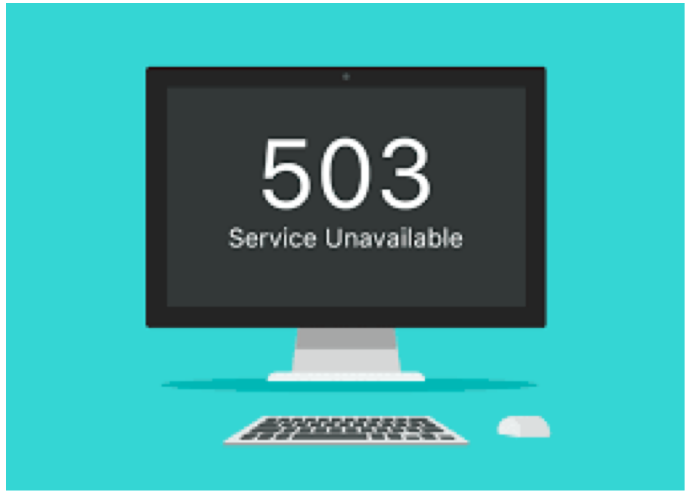
Handling invalid user input

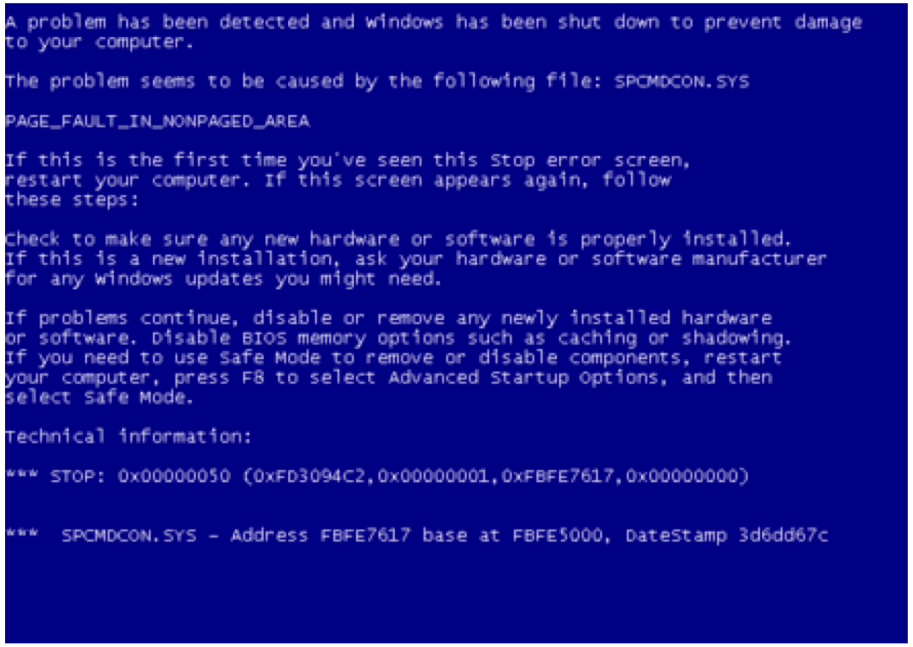
Produced by:
Dr. Siobhán Drohan
Mr. Colm Dunphy
Mr. Diarmuid O'Connor
Dr. Frank Walsh

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/


St. 404, Ghost Town
Looks like the page you're looking for
has been removed or doesn't exist

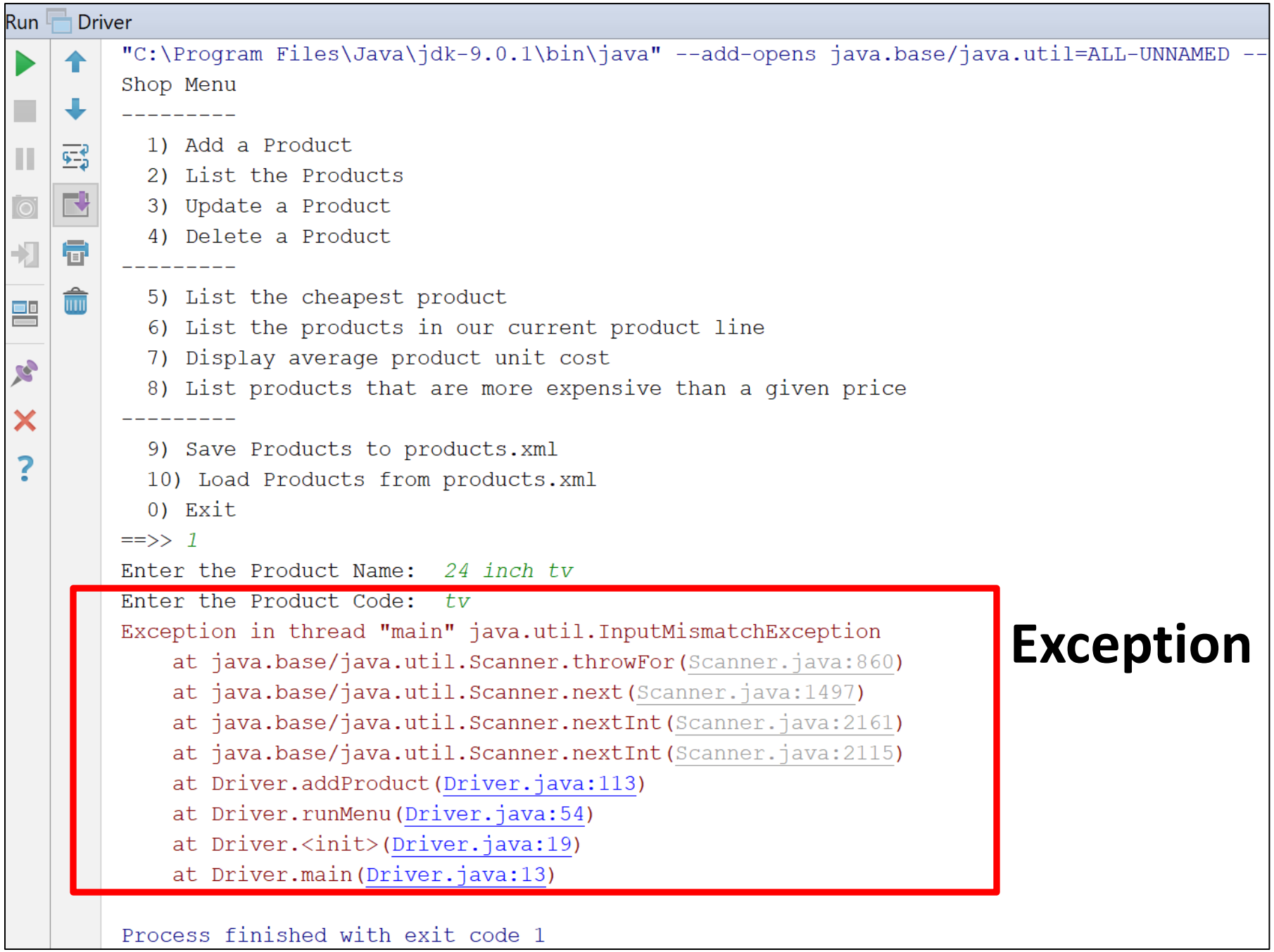

503
Service Unavailable


A problem has been detected and windows has been shut down to prevent damage
to your computer.
The problem seems to be caused by the following file: SPCMDCON.SYS
PAGE_FAULT_IN_NONPAGED_AREA
If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:
Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any windows updates you might need.
If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
Select Safe Mode.
Technical information:
*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x00000000)
*** SPCMDCON.SYS - Address FBFE7617 base at FBFE5000, DateStamp 3d6dd67c

Shop V5.0 (or any version)

- When testing data entry,
  did you try to enter a **String** instead of an **int**
  e.g. for the Product code?

- What happened?


Run
Driver
"C:\Program Files\Java\jdk-9.0.1\bin\java" --add-opens java.base/java.util=ALL-UNNAMED --
Shop Menu
----
1) Add a Product
2) List the Products
3) Update a Product
4) Delete a Product
----
5) List the cheapest product
6) List the products in our current product line
7) Display average product unit cost
8) List products that are more expensive than a given price
----
9) Save Products to products.xml
10) Load Products from products.xml
0) Exit
==>> 1
Enter the Product Name: 24 inch tv
Enter the Product Code: tv
Exception in thread "main" java.util.InputMismatchException
at java.base/java.util.Scanner.throwFor(Scanner.java:860)
at java.base/java.util.Scanner.next(Scanner.java:1497)
at java.base/java.util.Scanner.nextInt(Scanner.java:2161)
at java.base/java.util.Scanner.nextInt(Scanner.java:2115)
at Driver.addProduct(Driver.java:113)
at Driver.runMenu(Driver.java:54)
at Driver.<init>(Driver.java:19)
at Driver.main(Driver.java:13)
Process finished with exit code 1

Exception

Code

```
109         input.nextLine();
110         System.out.print("Enter the Product Name:  ");
111         String productName = input.nextLine();
112         System.out.print("Enter the Product Code:  ");
113         int productCode = input.nextInt();
114         System.out.print("Enter the Unit Cost:  ");
115         double unitCost = input.nextDouble();
116         System.out.print("Is this product in your current line (y/n): ");
117         char currentProduct = input.next().charAt(0);
118         boolean inCurrentProductLine = false;
119         if ((currentProduct == 'y') || (currentProduct == 'Y'))
120             inCurrentProductLine = true;
121
122         store.add(new Product(productName, productCode, unitCost, inCurrentProductLine));
123     }
```

```
Enter the Product Name:  24 inch tv
Enter the Product Code:  tv
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:860)
    at java.base/java.util.Scanner.next(Scanner.java:1497)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2161)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2115)
    at Driver.addProduct(Driver.java:113)
    at Driver.runMenu(Driver.java:54)
    at Driver.<init>(Driver.java:19)
    at Driver.main(Driver.java:13)

Process finished with exit code 1
```

Exception

thrown when reading an integer

Output

Shop V5.0 (or any version)

- The following code caused a runtime error...

```
int productCode = input.nextInt();
```

- This is called a **runtime exception**.

- How do we fix this?
- How do we stop the program from **crash**ing?

What are **Exceptions**?

- An Exception is an object
  that signals that some **unusual condition** has occurred
  while the program is executing.

- Exceptions are intended to be *detected* and *handled*,
  so that the program can continue in a sensible way
  if at all possible.

- Java has many **predefined Exception objects**.

When an exception occurs…

…the normal flow of execution is disrupted
and transferred to code,
which can handle the exception condition.

The exception mechanism is a lot cleaner
than having to check an error value
after every method call that could potentially fail.

RuntimeException…

- is a subclass of the Exception **class**.
- encompasses all exceptions which can ordinarily happen at run-time.
- these exceptions can be *thrown*
  by any java statement or a method call.
- can be avoided through good programming practices!

RuntimeException	Example Causes
ArithmeticException	Can be caused by dividing by zero.
ArrayIndexOutOfBoundsException	Referencing an array index number of 7 when only 5 exist in the array.
NullPointerException	Trying to access an object that has no memory allocated yet.

Catching Exceptions - **handlers**

- **Catch**ing an exception means
  declaring that you can handle exceptions of a particular class
  from a particular block of code.

- You specify the block of code
  and then provide **handlers** for various classes of exception.

- If an exception occurs
  then execution transfers to the corresponding piece of handler code.

St. 404, Ghost Town

Looks like the page you're looking for has been removed or doesn't exist

try and catch

To catch exceptions, you surround a block of code with a "try, catch" statement.

```
try{
    // The try clause is the piece of code which you want to try to execute.
    // it contains statements in which an exception could be raised
}
catch (Exception e){
    // The catch clauses are the handlers for the various exceptions.
    // it contains code to handle Exception and recover
}
```

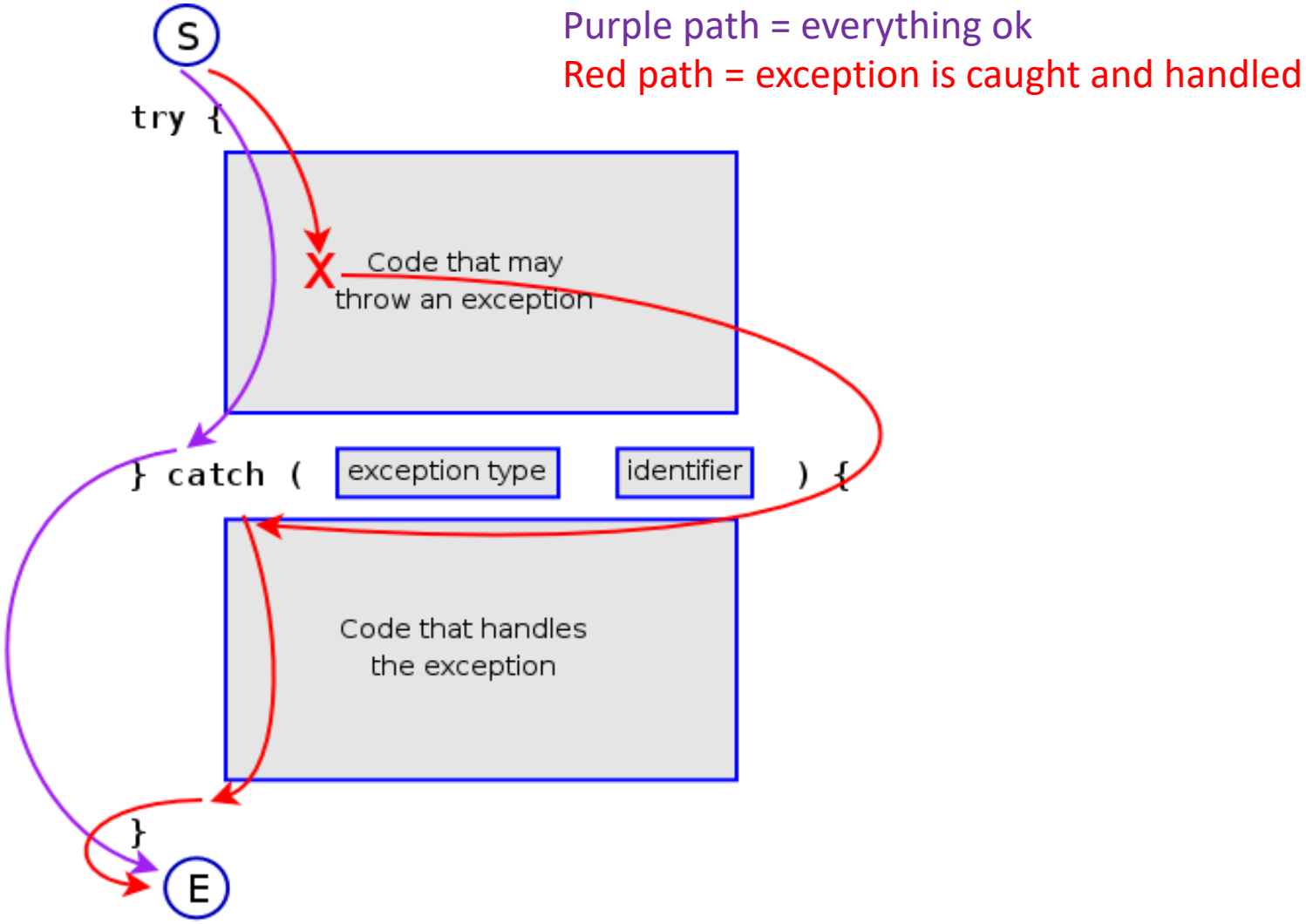
try and catch - **example**

```
try{
    myMethod();
}
catch (Exception e){
    System.err.println("Caught Exception:  " + e)
}
```

The parameter ***e*** is of type **Exception.**

We can use ***e*** to print out
what exception occurred.

Flow of control in Exception Handing



Returning to our **ShopV5.0**


```
int productCode = 0;

try {
      System.out.print("Enter the product code: ");
      productCode = input.nextInt();
}
catch (Exception e) {
       input.nextLine();   //swallows Scanner bug
       System.out.println("Number expected - you entered  text");
}
```

Improve – **loop** until input valid

```
int productCode = 0;
boolean goodInput = false;     //Loop  Control Variable

while (! goodInput ) {
    try {
            System.out.print("Enter the product code: ");
            productCode = input.nextInt();
            goodInput = true;
    }
    catch (Exception e) {
            input.nextLine();  //swallows Scanner bug
            System.out.println("Num expected - you entered  text");
    }
}
```

Same but using a **do...while** loop

```
int productCode = 0;
boolean goodInput = false;     //Loop  Control Variable

do {
    try {
        System.out.print("Enter the product code: ");
        productCode = input.nextInt();
        goodInput = true;
    }
    catch (Exception e) {
        input.nextLine();  //swallows Scanner bug
        System.out.println("Num expected - you entered  text");
    }
}  while (!goodInput);
```

Shop V5.0 (or any version)

- We have just seen how to stop this code from causing a runtime exception...

```
int productCode = input.nextInt();
```

- We should ideally
  - take this exception handling approach when reading in **any** numeric types.

Summary

- Crash v Exceptions
- Detect and Handle
  - Enables program to continue
- Java's predefined Exception objects
- try / catch block
- Introduction to
  - do while loop
    - Always runs once
    - Condition is test at the end

Any Questions?