Input & Output in Java

Standard I/O
Exception Handling

Java I/O: Generic & Complex

- Java runs on a huge variety of plaforms
- to accomplish this, a Java Virtual Machine (JVM) is written for every type of hardware
- the JVM handles details of I/O specific to each system

- output examples
  - monitor
  - phone display
  - audio/speaker
- input examples
  - keyboard
  - touchscreen
  - file
  - audio/voice

Standard Output

- for most systems, the JVM is configured to allow for simple text output
- this is part of the standard Java package
  - if you are writing a Java program, it should be automatically available

```
class StandardOutputTest
{
  public static void main(String[] args)
  {
    System.out.println("Java standard output.");
  }
}
```

In.class for Simplified Input

- you may have been using the In.class for input
- contains several helper routines which hide the complexity of Java input from programmer
- for different input (e.g., files) need to start to use a more generic input method

```
class StandardOutputTest
{
  public static void main(String[] args)
  {
    String name;
    int age;
    name = In.getString();
    age = In.getInt();
  }
}
```

Input Stream

- our default input method will be the keyboard
- also the default for many systems (e.g., PCs)
  - called System.in in most cases
  - System.out is generally a text box on screen
  - need to import standard input libraries

```
import java.io.*;
class StandardInputTest
{
  public static void main(String[] args)
  {
    InputStreamReader inStream = new InputStreamReader(System.in);
  }
}
```

Buffered Reader

- input data must be temporarily stored before being passed along to its final destination
- a buffer is a section of RAM set aside for this purpose
- it will hold all of the characters typed until the enter key is pressed

```
import java.io.*;
class StandardInputTest
{
  public static void main(String[] args)
  {
    InputStreamReader inStream = new InputStreamReader(System.in);
    BufferedReader bufRead = new BufferedReader(inStream);
  }
}
```

Exception Handling

- when a program is asked to perform an action, it generally assumes such an action is possible
- if a situation occurs where the action is not possible, the program will <u>throw an exception</u>
- for example
  - dividing by zero
  - array index out of bounds
  - reading from an empty buffer
- an unhandled exception will crash a program

try-catch Block

- when the failure of an action is outside of our control, we must include a try-catch block, looking for an IOException
  - some can be detected using code, and so are optional (e.g., divide by zero, if x == 0 ...)

```
InputStreamReader inStream = new InputStreamReader(System.in);
BufferedReader bufRead = new BufferedReader(inStream);

try
{
   String firstName = bufRead.readLine();
}
catch (IOException err)
{
   System.out.println("Error reading line");
}
```

Parsing Data

- all buffered data is initially a collection of characters assembled into one string per line
- to extract different data types, it is necessary to parse the string for the desired data

```
System.out.println("Please Enter The Year You Were Born: ");
String yearString = bufRead.readLine();

System.out.println("Please Enter Your Bank Balance: ");
String balanceString = bufRead.readLine();

int year = Integer.parseInt(yearString);
double balance = Double.parseDouble(balanceString);
```

Parsing Data requires try-catch

- the buffered reader may receive invalid data
- the parsing routines use the buffered reader
- therefore, the parsing activity may also be invalid, and requires a try-catch

```
try {
   System.out.println("Please Enter The Year You Were Born: ");
   String yearString = bufRead.readLine();
   int year = Integer.parseInt(yearString);
}
catch(NumberFormatException err) {
   System.out.println("Error Converting Number");
}
```

```java
import java.io.*;
class StandardInputTest
{
  public static void main(String[] args)
  {
    InputStreamReader inStream = new InputStreamReader(System.in);
    BufferedReader bufRead = new BufferedReader(inStream);

    try
    {
      System.out.println("Please Enter Your First Name: ");
      String firstName = bufRead.readLine();

      System.out.println("Please Enter The Year You Were Born: ");
      String yearString = bufRead.readLine();

      System.out.println("Please Enter Your Bank Balance: ");
      String balanceString = bufRead.readLine();

      int year = Integer.parseInt(yearString);
      double balance = Double.parseDouble(balanceString);
    }
    catch (IOException err)
    {
      System.out.println("Error reading line");
    }
    catch(NumberFormatException err) {
      System.out.println("Error Converting Number");
    }
  }
}
```