Ensuring reliability

- We do our best to ensure program correctness through a rigorous testing and debugging process
- To ensure reliability, we must anticipate conditions that could cause problems, and try to deal with them before the problems occur
- In Java, we have exception handling, a powerful tool for ensuring program reliability

Exceptions

- An error condition that occurs during program runtime is called an ***exception***
- Exceptions are represented by exception objects, which are generated (***thrown***) in response to error conditions
- Java includes a rich set of routines for dealing with such circumstances: this is known as ***exception handling***

Catching exceptions: try/catch block

- Consists of try block followed by one or more catch blocks
  - try block: encloses code that might throw an exception
  - catch block(s) deal with any exception(s) thrown

Syntax for try/catch block

```
try {
      // code that may throw an exception
    } catch (ExceptionType parameterName) {
      // code that handles an exception thrown
      // in the try block
}
```

Catching Exceptions

- Statements in the **try** block are executed in sequence.
- If no error occurs then no exception is thrown, all statements in the try block are executed and the catch block(s) will be skipped

Catching Exceptions

- Statements in the **try** block are executed in sequence.
- If no error occurs then no exception is thrown, all statements in the try block are executed and the catch block(s) will be skipped

Example

```
int value; // number to be supplied by user
String input; // value read from keyboard
Scanner kb = new Scanner (System.in);

System.out.print (“Enter a number: ”);
input = kb.nextLine();

try {
        value = Integer.parseInt(input);
    } catch (NumberFormatException e) {
        System.out.println (input + “is not valid.\n” +
                “Please enter digits only.”);
}
```

Example

```
public int getNumData (String prompt, int upper, int lower) {
  String inputStr;
  Scanner kb = new Scanner(System.in);
  int  num;
  while (true) {
      System.out.print(prompt);
      inputStr  = kb.nextLine();
      try {
          num = Integer.parseInt(inputStr);
          if (num < lower || num > upper)
              throw new RuntimeException("Input out of bounds");
          return num; // input okay so return the value & exit, ending loop
      } catch (NumberFormatException e) {
         System.out.println(inputStr + " is invalid\n" + "Please enter digits only");
     }  catch (RuntimeException e) {
          System.out.println(e.getMessage());
     } // end catch block
   } // end while loop
 } // end method
```