Overview of Java's Support for Packages & GC

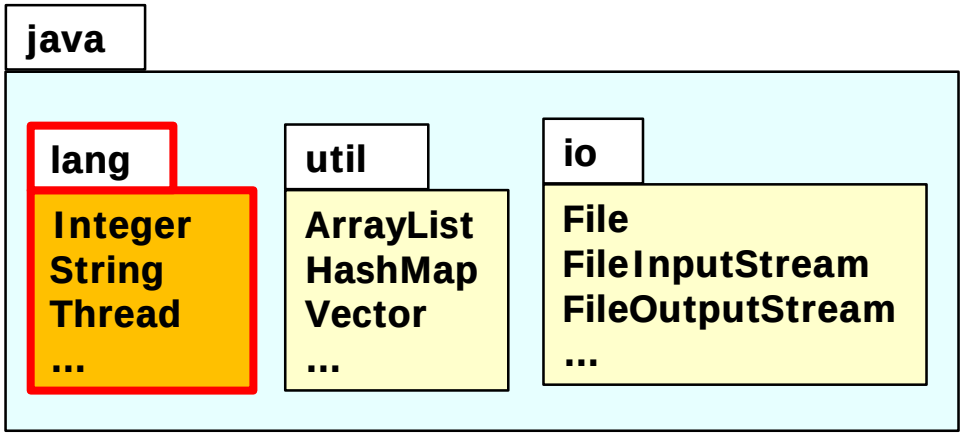
Overview of Java's Support for Packages & GC

- Classes & interfaces can be grouped into packages

See docs.oracle.com/javase/tutorial/java/concepts/package.html

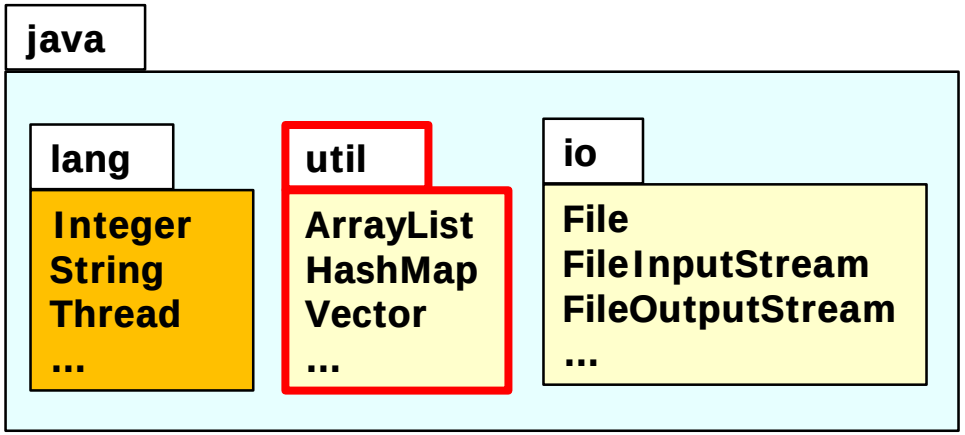
Overview of Java's Support for Packages & GC

- Classes & interfaces can be grouped into packages, e.g.
  - java.lang contains classes fundamental to design of Java



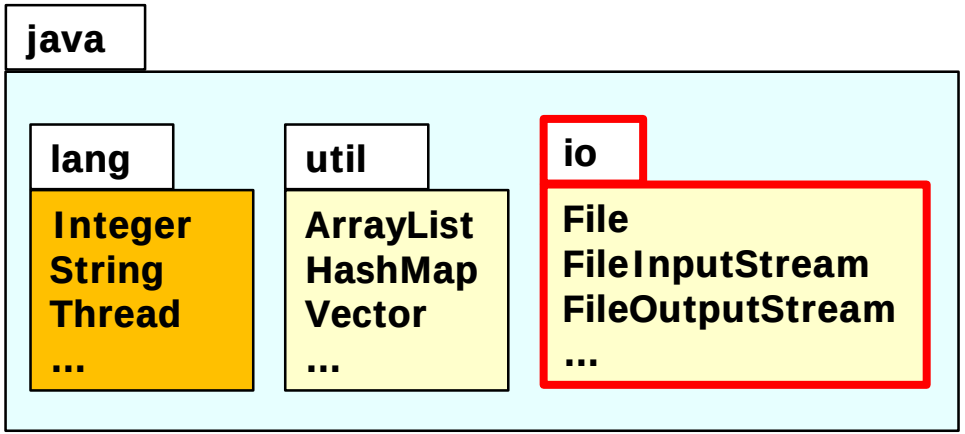
Overview of Java's Support for Packages & GC

- Classes & interfaces can be grouped into packages, e.g.
  - java.lang contains classes fundamental to design of Java
  - java.util contains a collection of common reusable ADTs



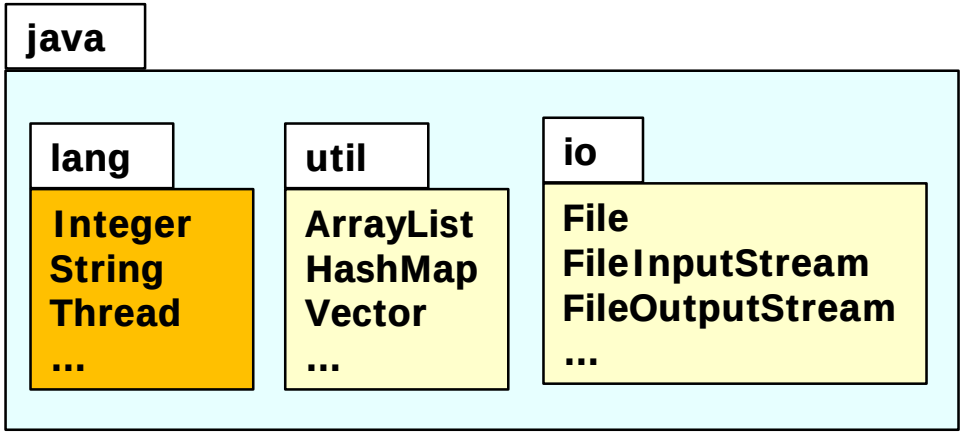
Overview of Java's Support for Packages & GC

- Classes & interfaces can be grouped into packages, e.g.
  - java.lang contains classes fundamental to design of Java
  - java.util contains a collection of common reusable ADTs
  - java.io contains classes that provide operations on files



Overview of Java's Support for Packages & GC

- Classes & interfaces can be grouped into packages



Packages help manage large projects by avoiding collisions for common names

Overview of Java's Support for Packages & GC

- Java's garbage collector automatically reclaims & recycles memory that is not in use by a program

See en.wikipedia.org/wiki/Garbage_collection_(computer_science)

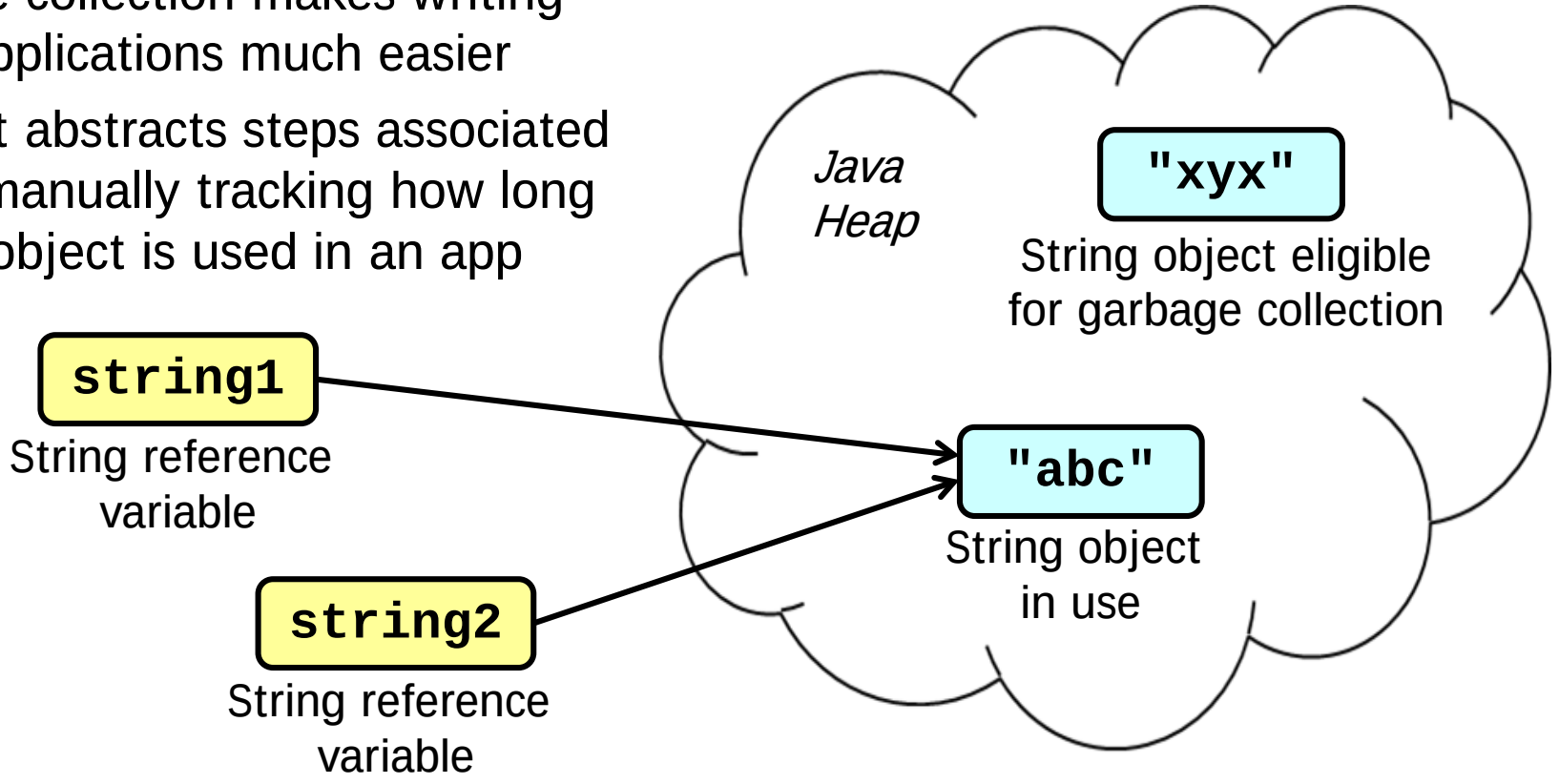
Overview of Java's Support for Packages & GC

- Garbage collection makes writing many applications much easier

See en.wikipedia.org/wiki/Garbage_collection_(computer_science)#Advantages

Overview of Java's Support for Packages & GC

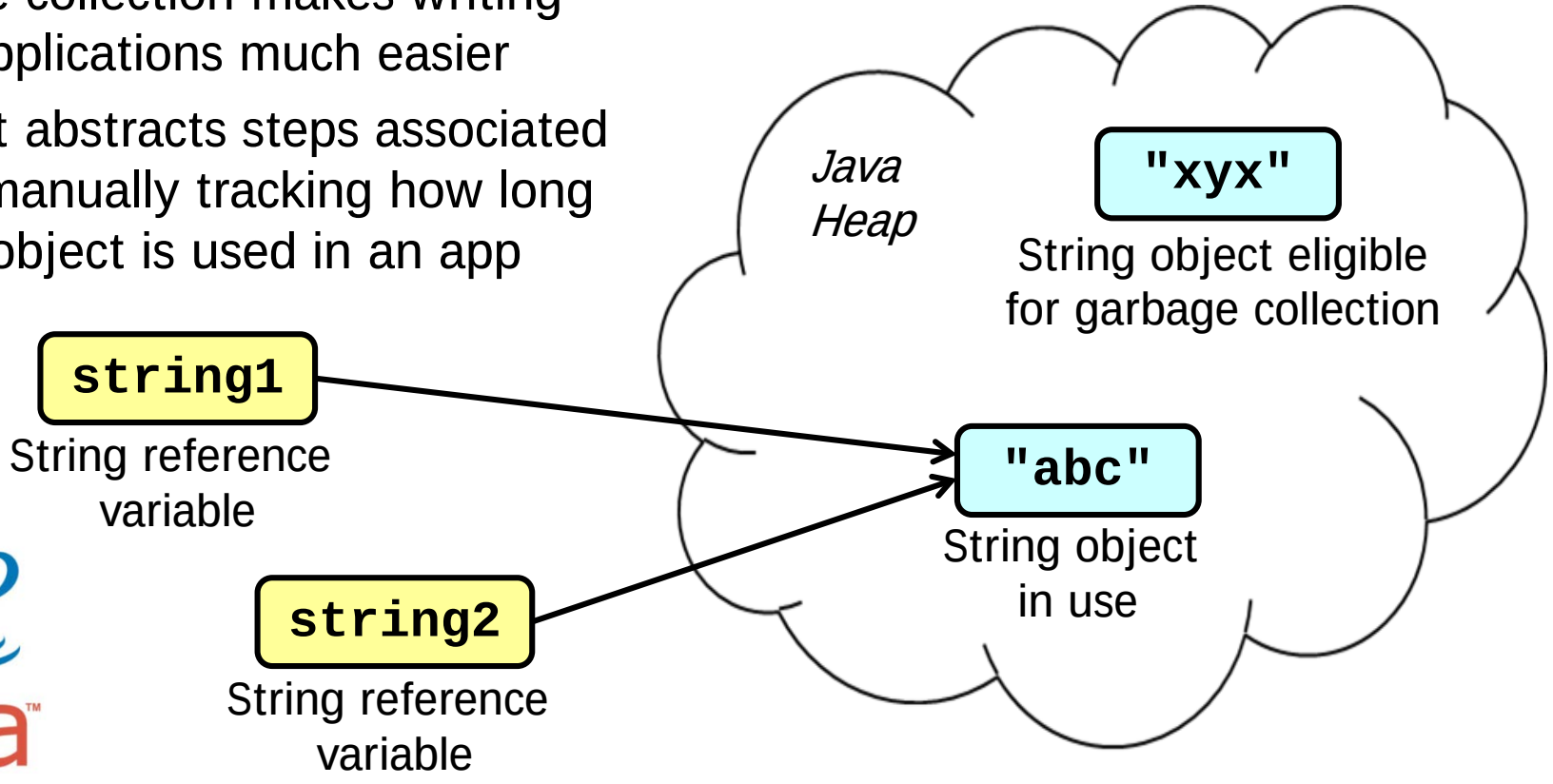
- Garbage collection makes writing many applications much easier
  - e.g., it abstracts steps associated with manually tracking how long each object is used in an app



Not all object-oriented languages support garbage collection...

Overview of Java's Support for Packages & GC

- Garbage collection makes writing many applications much easier
  - e.g., it abstracts steps associated with manually tracking how long each object is used in an app



... but garbage collection is an essential feature in Java

Overview of Java's Support for Packages & GC

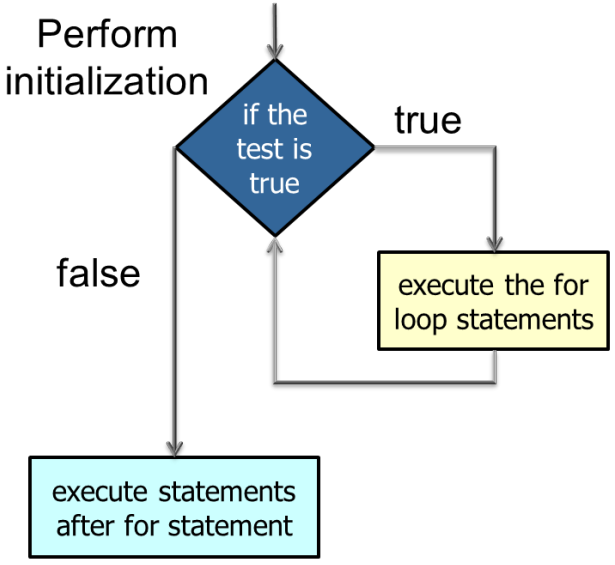
- There are both pros & cons of Java garbage collection

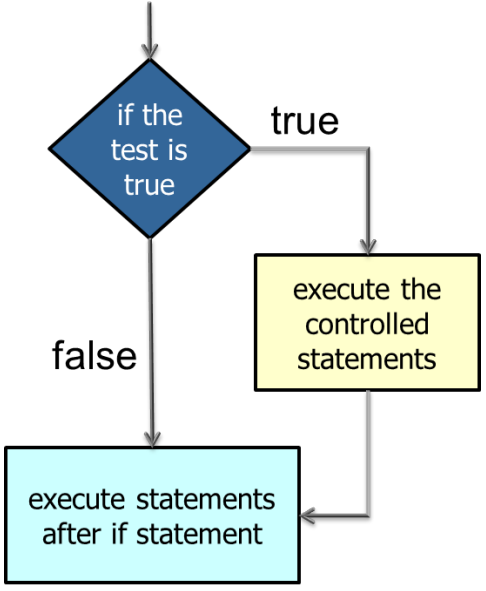
See blog.icodejava.com/574/all-about-java-garbage-collection-types-algorithms-advantages-and-disadvantages

Overview of Java's Support for Control Abstractions

Overview of Java's Support for Control Abstractions

- Java supports several control abstractions

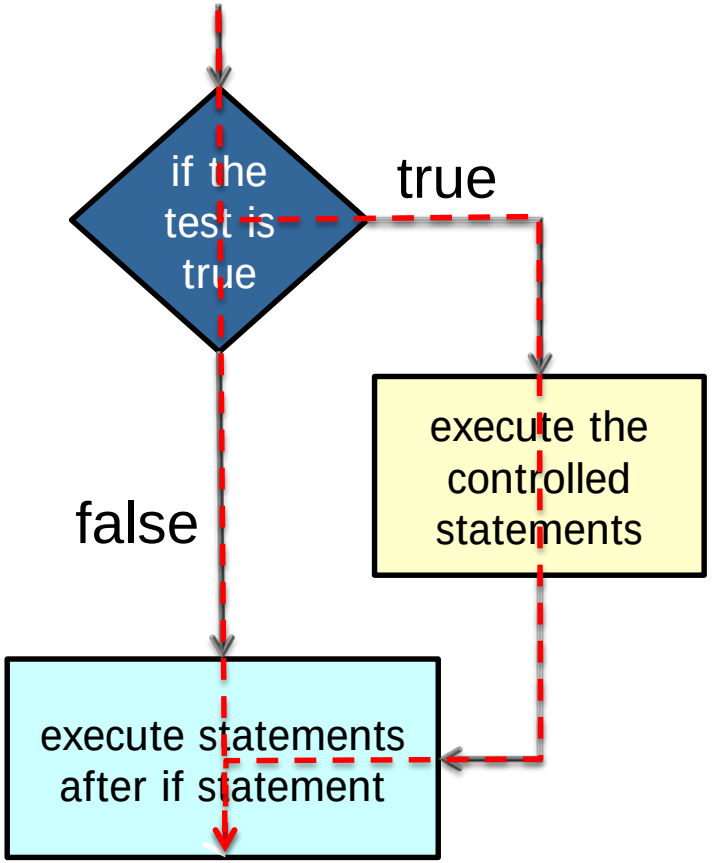




See en.wikipedia.org/wiki/Abstraction_(computer_science)#Control_abstraction

Overview of Java's Support for Control Abstractions

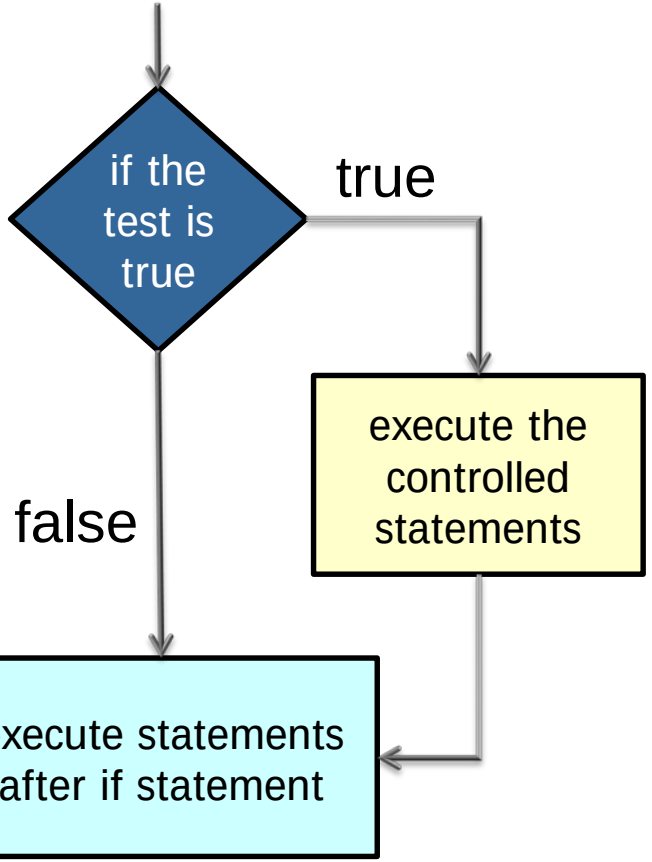
- Java conditional statements can be used to selectively alter program control flow



See en.wikipedia.org/wiki/Conditional_(computer_programming)

Overview of Java's Support for Control Abstractions

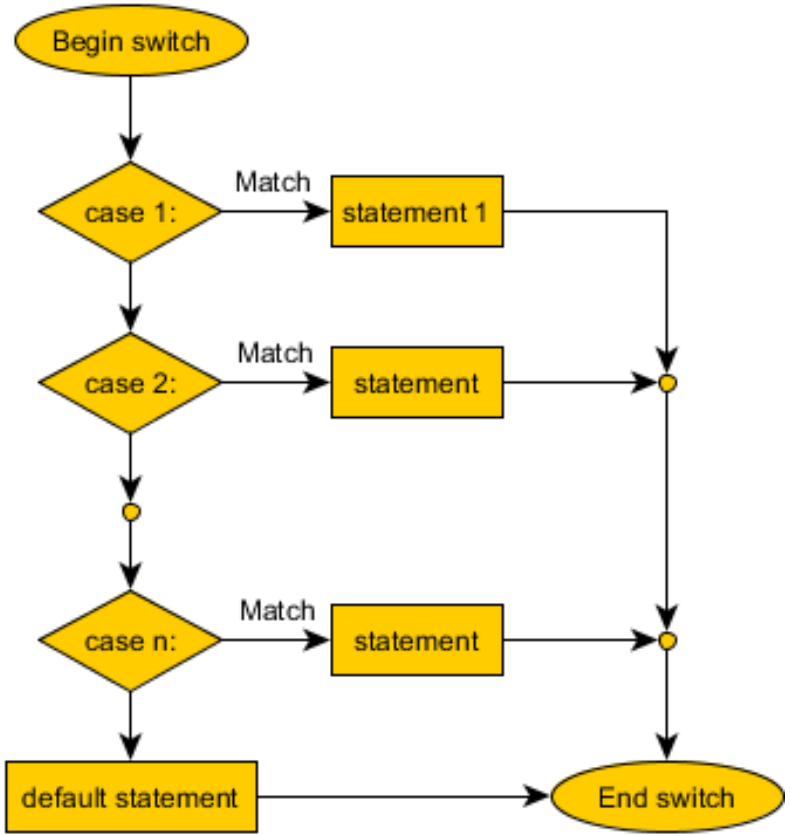
- Java conditional statements can be used to selectively alter program control flow, e.g.
  - if/else statement



See docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html

Overview of Java's Support for Control Abstractions

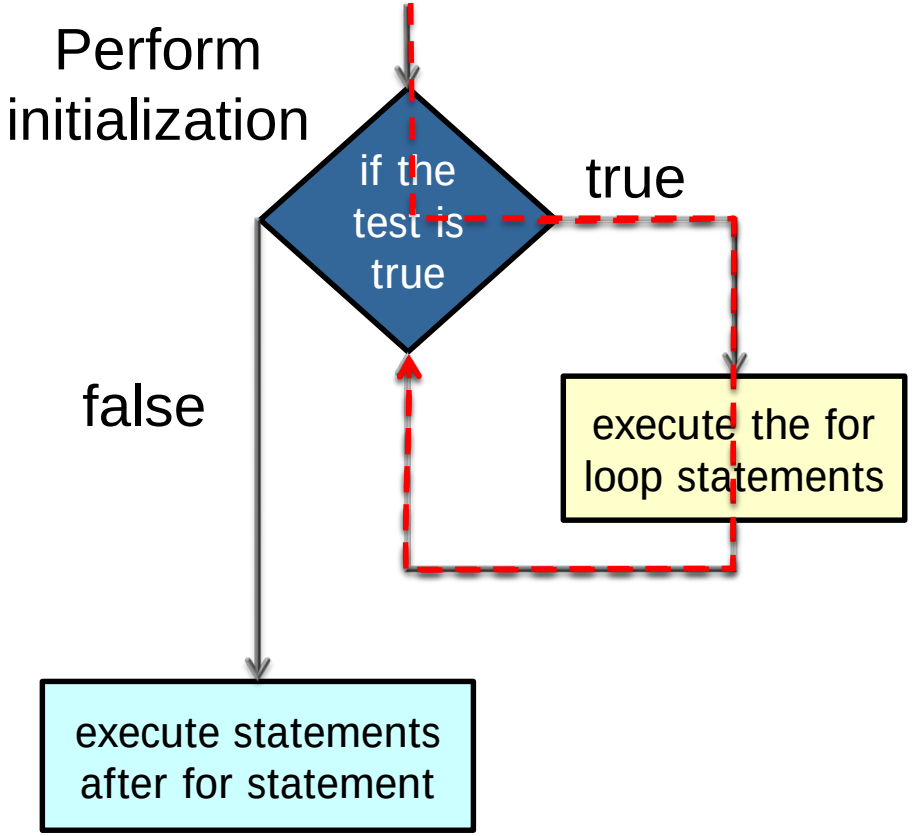
- Java conditional statements can be used to selectively alter program control flow, e.g.
  - if/else statement
  - switch statement



See docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html

Overview of Java's Support for Control Abstractions

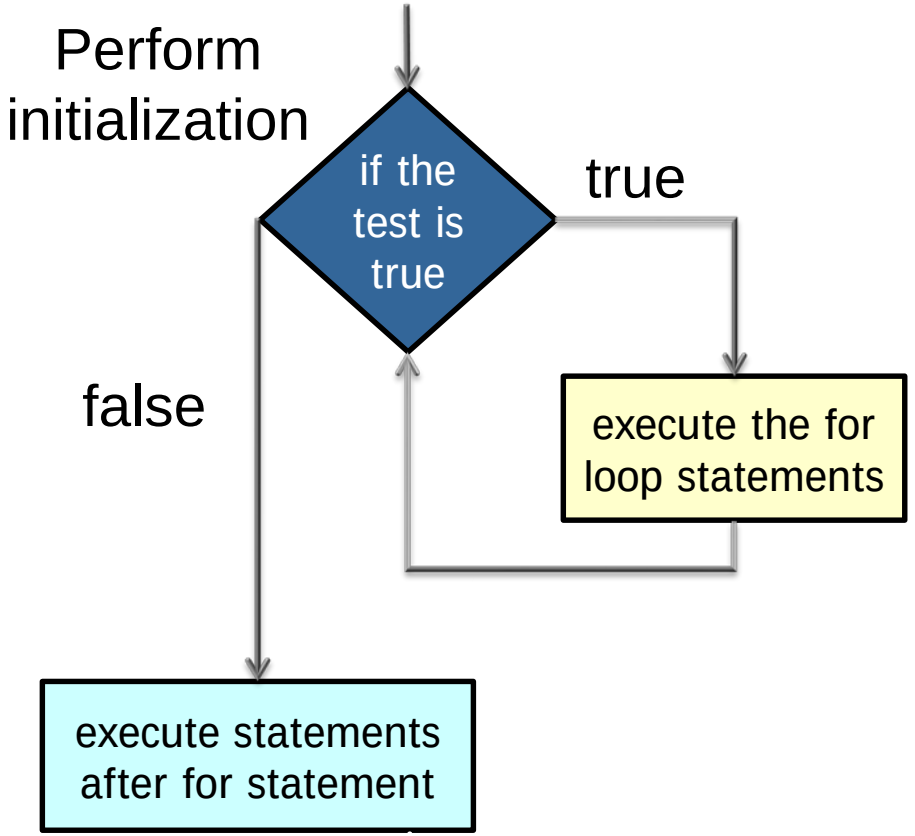
- Java iteration features enable repetition of a block of one or more statements



See en.wikipedia.org/wiki/Iteration#Computing

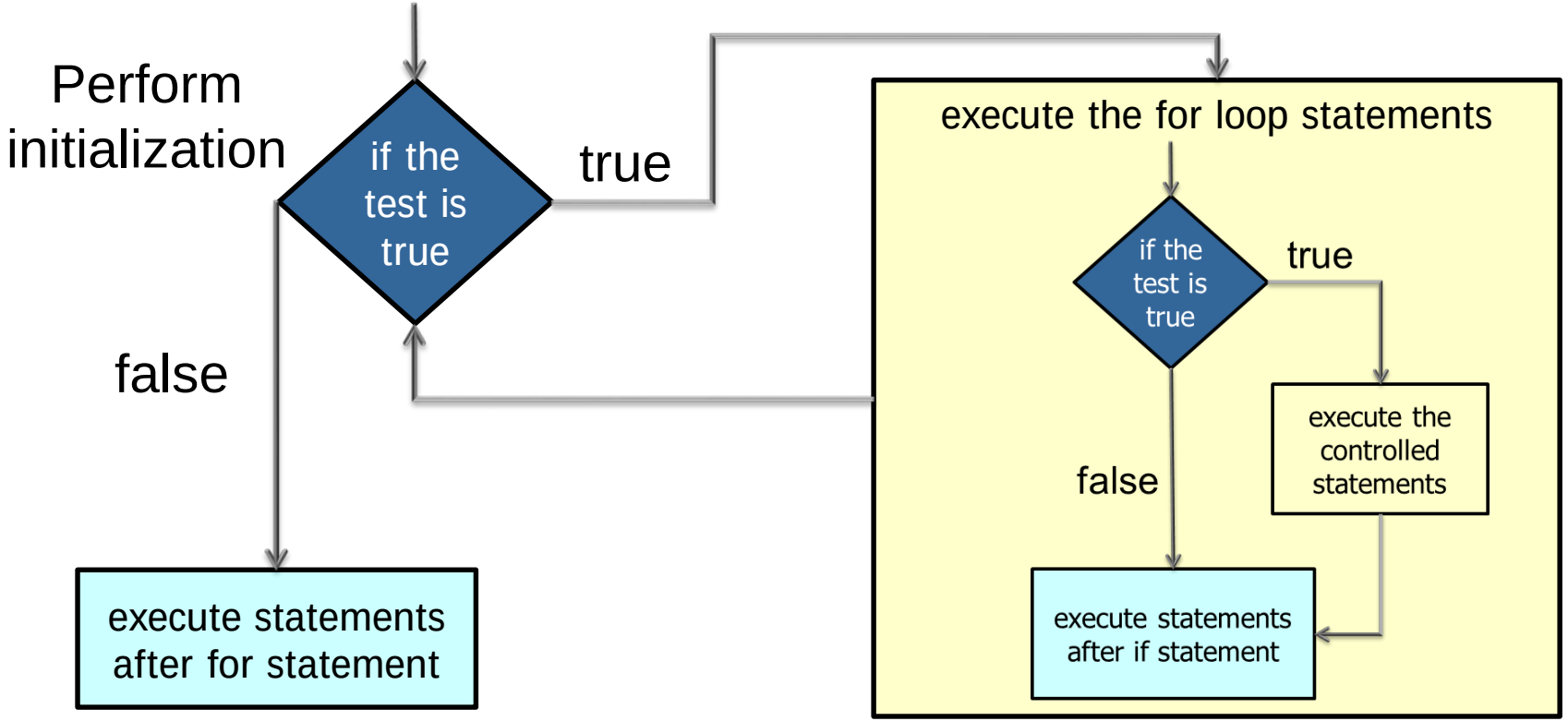
Overview of Java's Support for Control Abstractions

- Java iteration features enable repetition of a block of one or more statements
  - e.g., for loop, while loop, & do/while loop



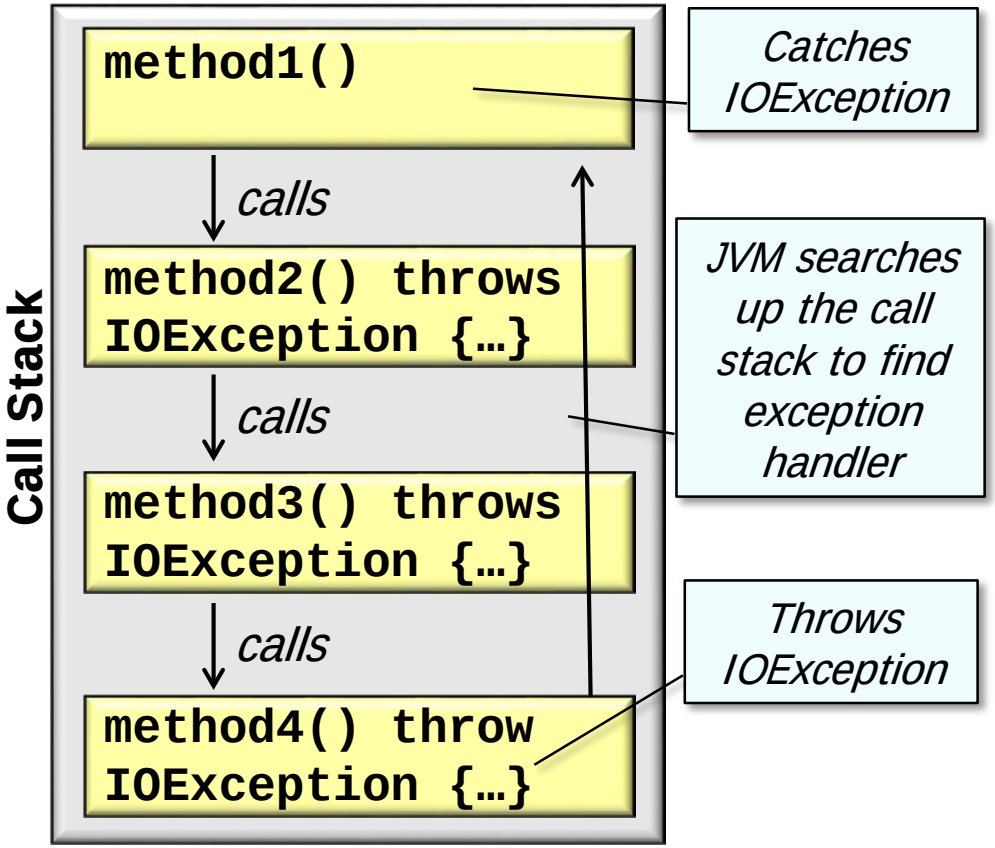
Overview of Java's Support for Control Abstractions

- Iteration features often combine with conditional statements in Java apps



Overview of Java's Support for Control Abstractions

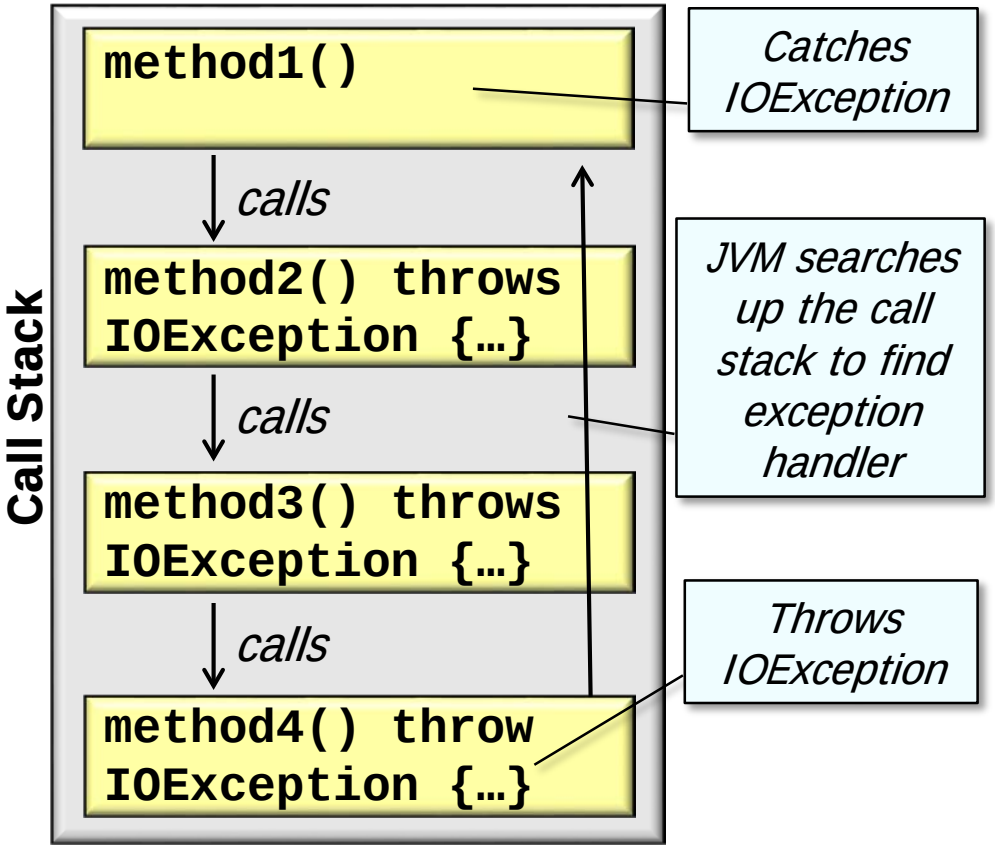
- Java exception handling separates "normal" app execution from "anomalous" app execution



See en.wikipedia.org/wiki/Exception_handling

Overview of Java's Support for Control Abstractions

- Java exception handling separates "normal" app execution from "anomalous" app execution



Exception handling makes Java apps more robust, understandable, & evolvable

Image Counter Example of Java Control Abstrations

Image Counter Example of Java Control Abstractions

- The countImages() method uses Java's if/else control abstraction to count images

```
int countImages(String pageUri,
                int depth) {
  if (depth > Options.instance()
      .maxDepth())
    return 0;
  else if (mUniqueUris
           .contains(pageUri))
    return 0;
  else {
    mUniqueUris.add(pageUri);

    return countImagesImpl
             (pageUri, depth);
  }
}
```

See ImageCounter/src/main/java/ImageCounter.java

Image Counter Example of Java Control Abstractions

- The countImagesImpl() method uses Java's for-each loop control abstraction to count images on (& accessible by) a page

```
int countImagesImpl(String uri,
                    int depth) {
  Document page = getStartPage(uri);

  int imagesInPage =
    getImagesOnPage(page).size();

  List<Integer> imagesInLinksList =
    crawlLinksInPage(page, depth);

  for (int c : imagesInLinksList)
    imagesInPage += c;

  return imagesInPage;
}
```

Image Counter Example of Java Control Abstractions

- The countImagesImpl() method uses Java's for-each loop control abstraction to count images on (& accessible by) a page

```
int countImagesImpl(String uri,
                    int depth) {
  Document page = getStartPage(uri);

  int imagesInPage =
    getImagesOnPage(page).size();

  return imagesInPage +
    crawlLinksInPage(page, depth)
      .stream()
      .mapToInt(Integer::intValue)
      .sum();
}
```

Explicit looping & "external iteration" is largely unnecessary in Java 8 (due to streams)

End of Overview of Java's Support for Data & Control Abstraction