Assertions

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Programming Concepts using Java

Week 7

Documenting and checking assumptions

- Functions may have constraints on the parameters

```
public static double myfn(double x){
  // Assume x >= 0
  ...
}
```

Documenting and checking assumptions

- Functions may have constraints on the parameters
- We could check the condition and throw an exception

```
public static double myfn(double x)
  throws IllegalArgumentException {
  // Assume x >= 0
  if (x < 0){
    throw new
          IllegalArgumentException("x < 0");
  }
}
```

Documenting and checking assumptions

- Functions may have constraints on the parameters
- We could check the condition and throw an exception
- What if `myfn` is only used internally by our own code
  - Flag errors during development, debugging
  - But diagnostic code should not trigger at run time
  - Performance, and other considerations

```
public static double myfn(double x)
  throws IllegalArgumentException {
  // Assume x >= 0
  if (x < 0){
    throw new
          IllegalArgumentException("x < 0");
  }
}
```

Documenting and checking assumptions

- Functions may have constraints on the parameters
- We could check the condition and throw an exception
- What if `myfn` is only used internally by our own code
  - Flag errors during development, debugging
  - But diagnostic code should not trigger at run time
  - Performance, and other considerations
- Instead, “assert” the property you assume to hold

```
public static double myfn(double x){
  assert x >= 0;
}
```

Assertions

- If assertion fails, code throws `AssertionError`

```
public static double myfn(double x){
  assert x >= 0;
}
```

Assertions

- If assertion fails, code throws `AssertionError`
- This should not be caught
  - Abort and print diagnostic information (stack trace)

```
public static double myfn(double x){
  assert x >= 0;
}
```

Assertions

- If assertion fails, code throws `AssertionError`
- This should not be caught
  - Abort and print diagnostic information (stack trace)
- Can provide additional information to be printed with diagnostic message

```
public static double myfn(double x){
  assert x >= 0 : x;
}
```

Enabling and disabling assertions

- Assertions are enabled or disabled at runtime
    - Does not require recompilation

Enabling and disabling assertions

- Assertions are enabled or disabled at runtime
  - Does not require recompilation
- Use the following flag to run with assertions enabled

```
java -enableassertions MyCode
```

Enabling and disabling assertions

- Assertions are enabled or disabled at runtime
  - Does not require recompilation
- Use the following flag to run with assertions enabled

```
java -enableassertions MyCode
```

- Can use `-ea` as abbreviation for `-enableassertions`

Enabling and disabling assertions

- Assertions are enabled or disabled at runtime
  - Does not require recompilation
- Use the following flag to run with assertions enabled

  ```
  java -enableassertions MyCode
  ```

- Can use `-ea` as abbreviation for `-enableassertions`
- Can selectively turn on assertions for a class

  ```
  java -ea:Myclass MyCode
  ```

Enabling and disabling assertions

- Assertions are enabled or disabled at runtime
    - Does not require recompilation
- Use the following flag to run with assertions enabled

```
java -enableassertions MyCode
```

- Can use `-ea` as abbreviation for `-enableassertions`
- Can selectively turn on assertions for a class

```
java -ea:Myclass MyCode
```

- . . . or a package

```
java -ea:in.ac.iitm.onlinedegree MyCode
```

Enabling and disabling assertions

- Assertions are enabled or disabled at runtime
  - Does not require recompilation
- Use the following flag to run with assertions enabled

```
java -enableassertions MyCode
```

- Can use `-ea` as abbreviation for `-enableassertions`
- Can selectively turn on assertions for a class

```
java -ea:Myclass MyCode
```

- . . . or a package

```
java -ea:in.ac.iitm.onlinedegree MyCode
```

- Similarly, disable assertions globally or selectively

```
java -disableassertions MyCode
java -da:MyClass MyCode
```

Enabling and disabling assertions

- Assertions are enabled or disabled at runtime
  - Does not require recompilation
- Use the following flag to run with assertions enabled

```
java -enableassertions MyCode
```

- Can use `-ea` as abbreviation for `-enableassertions`
- Can selectively turn on assertions for a class

```
java -ea:Myclass MyCode
```

- . . . or a package

```
java -ea:in.ac.iitm.onlinedegree MyCode
```

- Similarly, disable assertions globally or selectively

```
java -disableassertions MyCode
java -da:MyClass MyCode
```

- Can combine the two

```
java -ea in.ac.iitm.onlinedegree
     -da:MyClass MyCode
```

Enabling and disabling assertions

- Assertions are enabled or disabled at runtime
  - Does not require recompilation
- Use the following flag to run with assertions enabled

```
java -enableassertions MyCode
```

- Can use `-ea` as abbreviation for `-enableassertions`
- Can selectively turn on assertions for a class

```
java -ea:Myclass MyCode
```

- . . . or a package

```
java -ea:in.ac.iitm.onlinedegree MyCode
```

- Similarly, disable assertions globally or selectively

```
java -disableassertions MyCode
java -da:MyClass MyCode
```

- Can combine the two

```
java -ea in.ac.iitm.onlinedegree
     -da:MyClass MyCode
```

- Separate switch to enable assertions for system classes

```
java -enablesystemassertions MyCode
java -esa MyCode
```

Summary

- Assertion checks are supposed to flag fatal, unrecoverable errors
    - Do not `catch` them!
- If you need to flag the error and take corrective action, use exceptions instead
- Turned on only during development and testing
    - Not checked at run time after deployment