CMPT 276
Intro to Software Engineering

Assertions & Refactoring

Nick Sumner – Spring 2015

Assertions

- Assertions
  - provide a way of checking *invariants* (internal expectations)

Assertions

- Assertions
  - provide a way of checking *invariants* (internal expectations)

Check whether a condition is true.
Crash if it is not.

Assertions

- Assertions
  - provide a way of checking *invariants* (internal expectations)
  - In Java (run using -enableassertions):

```
private void setScore(int score) {
    assert score >= 0 && score <= 100
      : “Score of ” + score + “is out of range”;
```

Assertions

- Assertions
  - provide a way of checking *invariants* (internal expectations)
  - In Java (run using -enableassertions):

```
private void setScore(int score) {
    assert score >= 0 && score <= 100
      : “Score of ” + score + “is out of range”;
```

  - In C/C++ (enabled by default):

```
void setScore(int score) {
    assert(score >= 0 && score <= 100
           && “Score is out of range”);
```

    - Disable by compiling with -DNDEBUG

Assertions

- Why?

Why crash your own code?

Assertions

- Why?
  - Software with assertions has fewer defects

Assertions

- Why?
  - Software with assertions has fewer defects
  - Documents the software via the code

Assertions

- Why?
  - Software with assertions has fewer defects
  - Documents the software via the code
  - The defects that are there are easier to diagnose
    - Incorrect values are spotted close to the defect

Assertions

- Why?
  - Software with assertions has fewer defects
  - Documents the software via the code
  - The defects that are there are easier to diagnose
    - Incorrect values are spotted close to the defect

A little work up front makes your life easier!

Assertions vs Exceptions

- You might instead throw an exception:

```
public void setScore(int score) {
  If (score < 0 || score > 100) {
    throw new InvalidArgumentException(“Score of ”
            + score + “is out of range”);
```

Assertions vs Exceptions

- You might instead throw an exception:

```
public void setScore(int score) {
  If (score < 0 || score > 100) {
    throw new InvalidArgumentException("Score of "
            + score + "is out of range");
```

- Why use one vs the other?

Assertions vs Exceptions

- You might instead throw an exception:

```
public void setScore(int score) {
  If (score < 0 || score > 100) {
    throw new InvalidArgumentException("Score of "
            + score + "is out of range");
```

- Why use one vs the other?

Internal invariants vs. External expectations
(errors in your own code) vs. (errors in user input)
assertions vs. exceptions

Assertions vs Exceptions

- You might instead throw an exception:

```
public void setScore(int score) {
  If (score < 0 || score > 100) {
    throw new InvalidArgumentException("Score of "
            + score + "is out of range");
```

- Why use one vs the other?

Internal invariants vs. External expectations
(errors in your own code) vs. (errors in user input)
assertions vs. exceptions

Assertions indicate internal bugs.
Exceptions indicate misuse / corner cases.

Refactoring

- Refactoring
  - Restructuring existing code for better:
    - Readability
    - Flexibility
    - Testability
    - … bility

Refactoring

- Refactoring
  - Restructuring existing code for better:
    - Readability
    - Flexibility
    - Testability
    - … bility
  - Changes to code that do not affect functional behavior

Refactoring – examples from Martin Fowler

- Extract Explaining Variable:
  - http://refactoring.com/catalog/extractVariable.html

Refactoring – examples from Martin Fowler

- Extract Explaining Variable:
  - http://refactoring.com/catalog/extractVariable.html
- Split Temporary Variable:
  - http://refactoring.com/catalog/splitTemporaryVariable.html

Refactoring – examples from Martin Fowler

- Extract Explaining Variable:
  - http://refactoring.com/catalog/extractVariable.html
- Split Temporary Variable:
  - http://refactoring.com/catalog/splitTemporaryVariable.html
- Extract Method:
  - http://refactoring.com/catalog/extractMethod.html

Refactoring – examples from Martin Fowler

- Extract Explaining Variable:
  - http://refactoring.com/catalog/extractVariable.html
- Split Temporary Variable:
  - http://refactoring.com/catalog/splitTemporaryVariable.html
- Extract Method:
  - http://refactoring.com/catalog/extractMethod.html
- Replace Temp with Query
  - http://refactoring.com/catalog/replaceTempWithQuery.html

Refactoring

The canonical resource is http://www.refactoring.com