CS 2050
Computer Science II

Lesson 03

METROPOLITAN STATE UNIVERSITY℠
OF DENVER
LIVES TRANSFORMED

Agenda

- Instance vs. Class Variables
- Software Development Testing

METROPOLITAN STATE UNIVERSITY℠ OF DENVER
LIVES TRANSFORMED

Review

- Member variables:
  - variables defined within a class but outside functions
  - they can be class variables or instance variables
  - member variables are also called fields, attributes, or properties
  - access to member variables can be made more restrict using the private keyword

METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Review

- Methods:
  - functions defined within a class
  - they can be class methods or instance methods
  - constructors are special methods that are used to initialize your objects
  - methods have parameters and explicit return values (with the exception of constructors, they implicitly return a reference to the object created)

METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Review

- Constructors:
  - Special methods that are used during instantiation only
  - In Java, they have the same name of the class where they are defined
  - They are called when you use the new operator
  - They don’t explicitly return a value (because they actually return a reference to the object created)

METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Instance x Class Variables

- Instance variables:
  - There is one for each instance
  - In other words, each object has its own copy

METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Instance x Class Variables

- Class variables:
  - There is one for ALL instances of the class
  - In other words, objects share class variables
  - To define a class variables in Java add the static keyword

METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Example

- Consider a class called Egg
- It makes sense to define the following as instance variables:

```
private int size;
private boolean freeRange;
```


METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Example

- Consider a class called Egg
- It also makes sense to define the following (constants) as class variables:

```
private static final int SMALL      = 1;
private static final int MEDIUM     = 2;
private static final int LARGE      = 3;
private static final int EXTRA_LARGE = 4;
```


METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Example

- Constructors:
  - No parameters (size should be MEDIUM, freeRange false)
  - Accept freeRange, set size to MEDIUM
  - Accept size (validate, defaults to MEDIUM), freeRange false
  - Accepts size (validate, defaults to MEDIUM) and freeRange

METROPOLITAN STATE UNIVERSITY℠ OF DENVER
LIVES TRANSFORMED

Instance x Class Variables

- Class variables - when to use:
  - When defining constants
  - Constants in Java are defined using the keyword final
  - When you need to share information between instances:
    - For example: a global counter

METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Software Development Testing

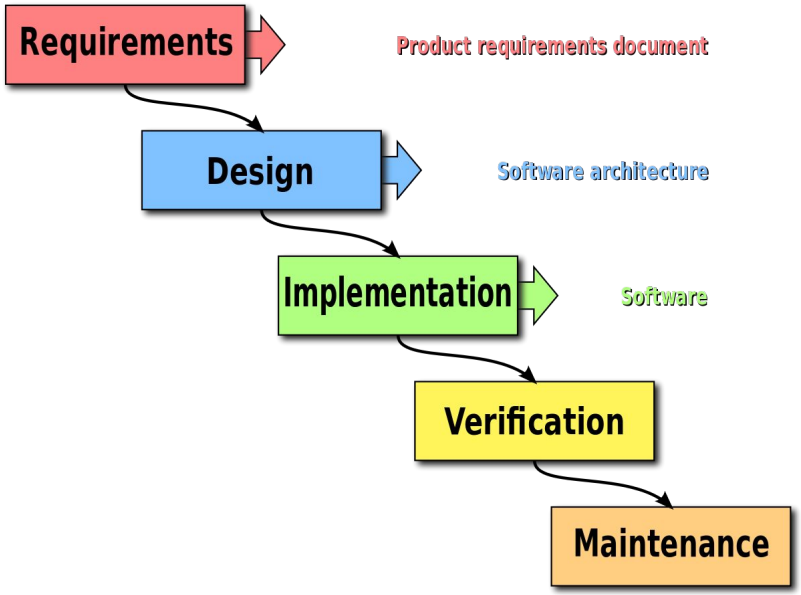
- Ariane 5 Failure (1996):
  - Rocket exploded just 40s after its lift-off
  - $7 billion dollar project
  - Caused by a 64-bit floating-point number converted to a 16-bit signed integer

Software Development Testing

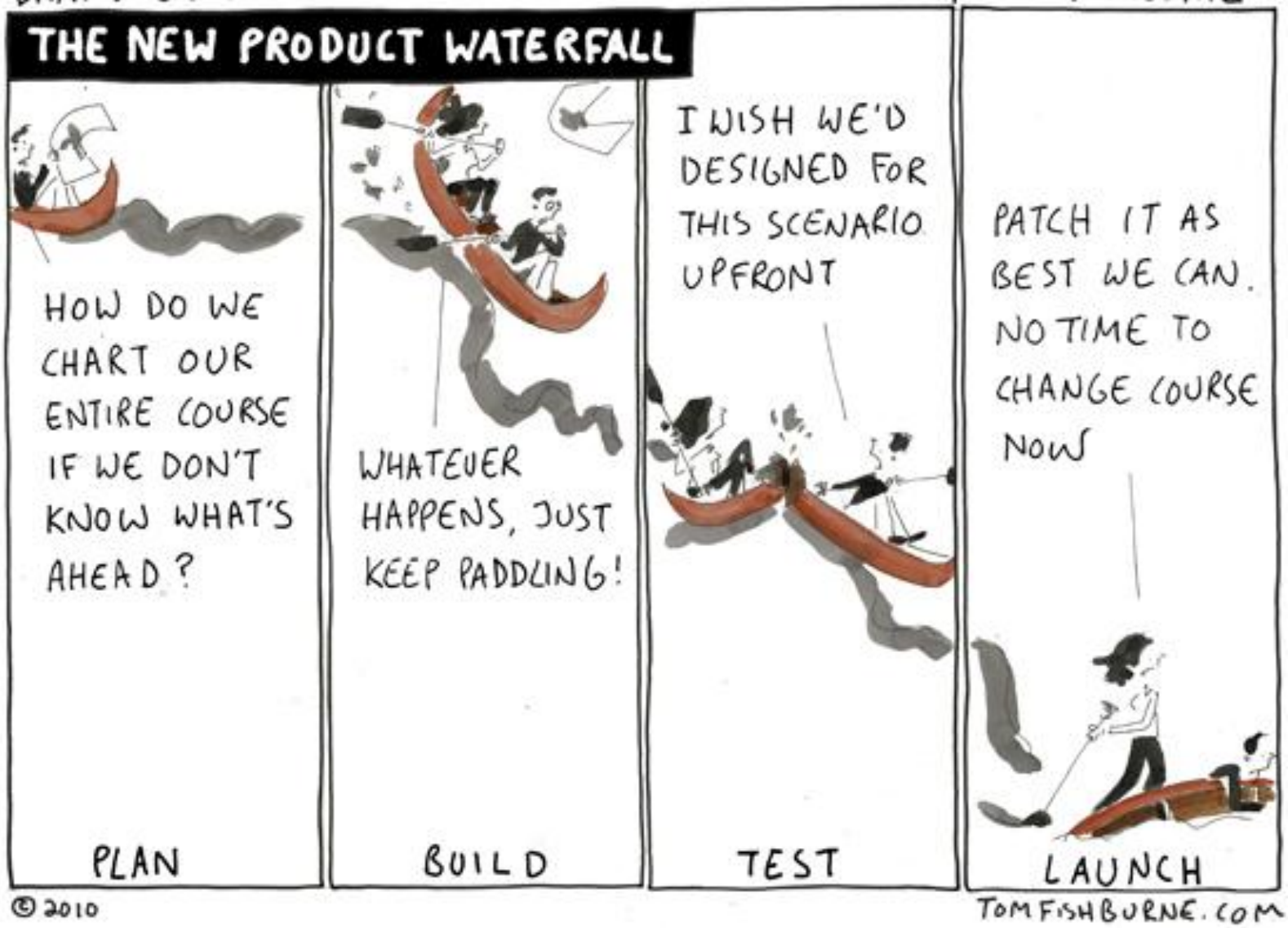
LIVE UPDATES **A major coding error is behind the app issue that appears to have caused inconsistencies in vote reporting at the caucuses, a source says**

Software Development Testing

- History:
  - Waterfall Model



METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED



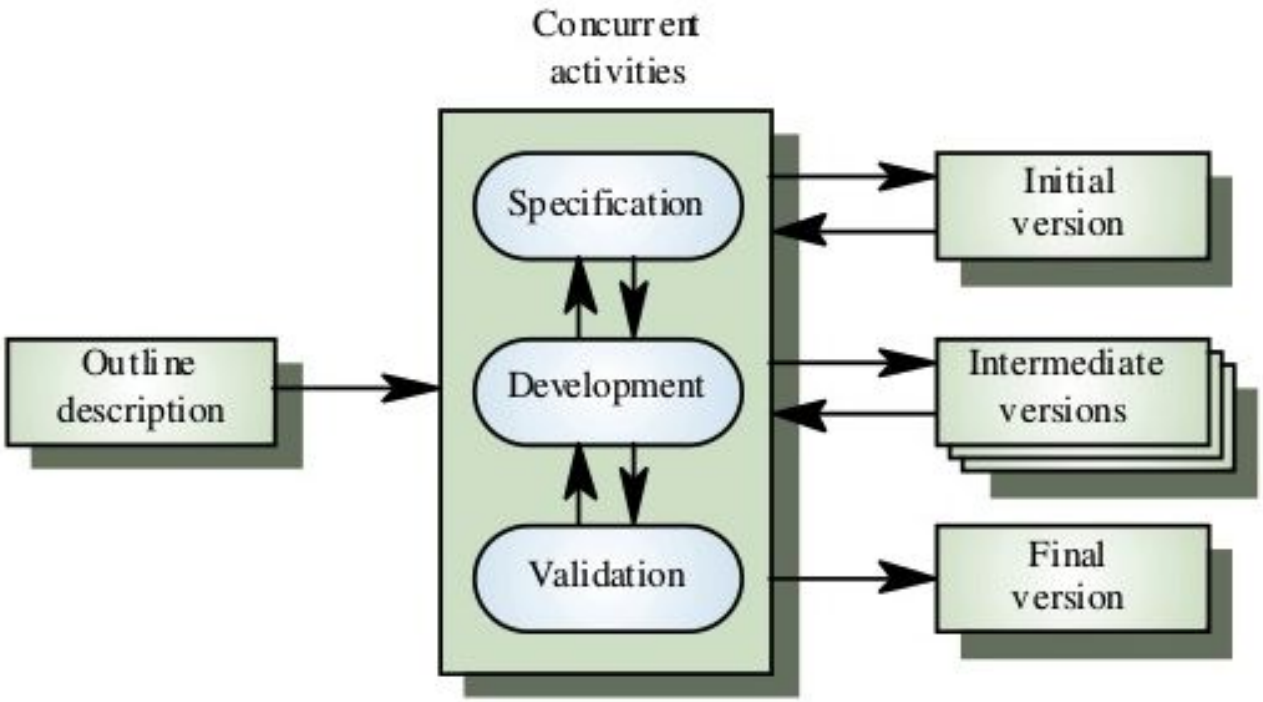
METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Software Development Testing

- Deliver working software incrementally in short iterations
- Get working software in front of real users as soon as possible
- Software is delivered in multiple versions until the product reaches user satisfaction
- Software is constantly being changed and updated:
  - Crucial that each version passes validation procedures

METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

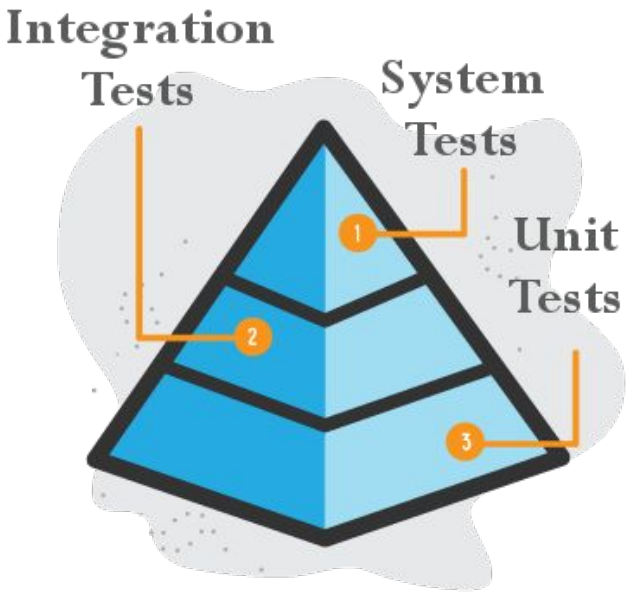
Software Development Testing



METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Software Development Testing

- Testing Levels:



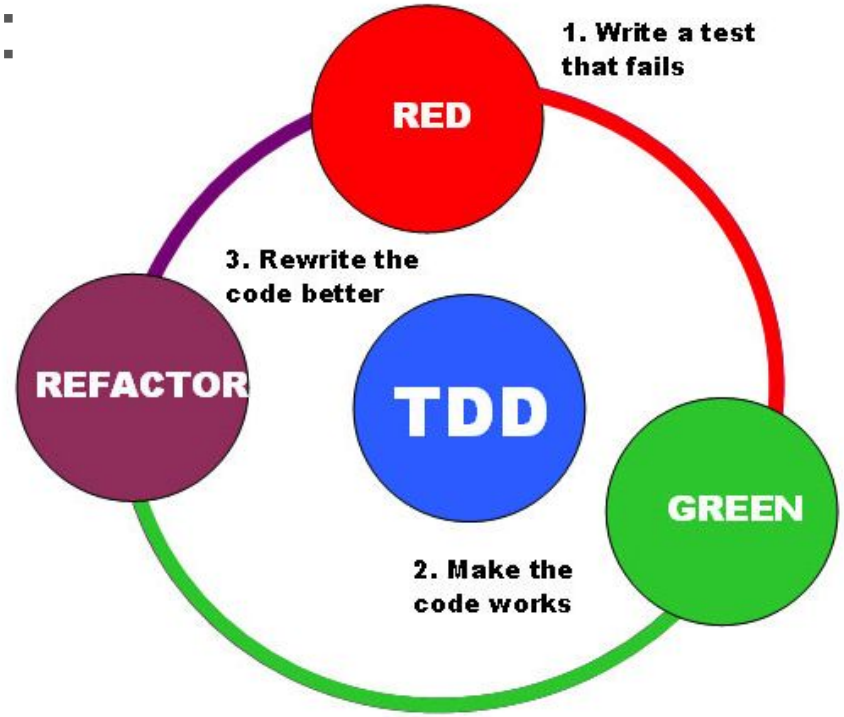
Software Development Testing

- Unit Testing:
  - Individual units of software code (function, class, etc.) are tested
  - The goal is to isolate each part of the program and show that the individual parts are correct
  - Unit testing can be automated using testing frameworks (JUnit in Java, unittest in Python, etc.)

METROPOLITAN STATE UNIVERSITY℠ OF DENVER
LIVES TRANSFORMED

Software Development Testing

- Unit Testing:
  - Best Practices:



Software Development Testing

- Best Practices:
  - Give your tests names as descriptive as possible
  - Describe the inputs and expected outputs of your test
  - Use the scientific method:
    - Formulate hypothesis
    - Test the hypothesis
  - Make your tests as atomic as possible
  - Avoid test interdependence

METROPOLITAN STATE UNIVERSITY OF DENVER
LIVES TRANSFORMED

Software Development Testing

- Unit Testing:
  - Floating-point Approximation Errors:

```
double a = 0;
for (int i = 0; i < 10; i++)
    a += 0.1;

double b = 1;

System.out.println("Comparison using equality operator:");
if (a == b)
    System.out.println("a == b");
else
    System.out.println("a != b");
```

Software Development Testing

- Unit Testing:
  - Floating-point Approximation Errors:

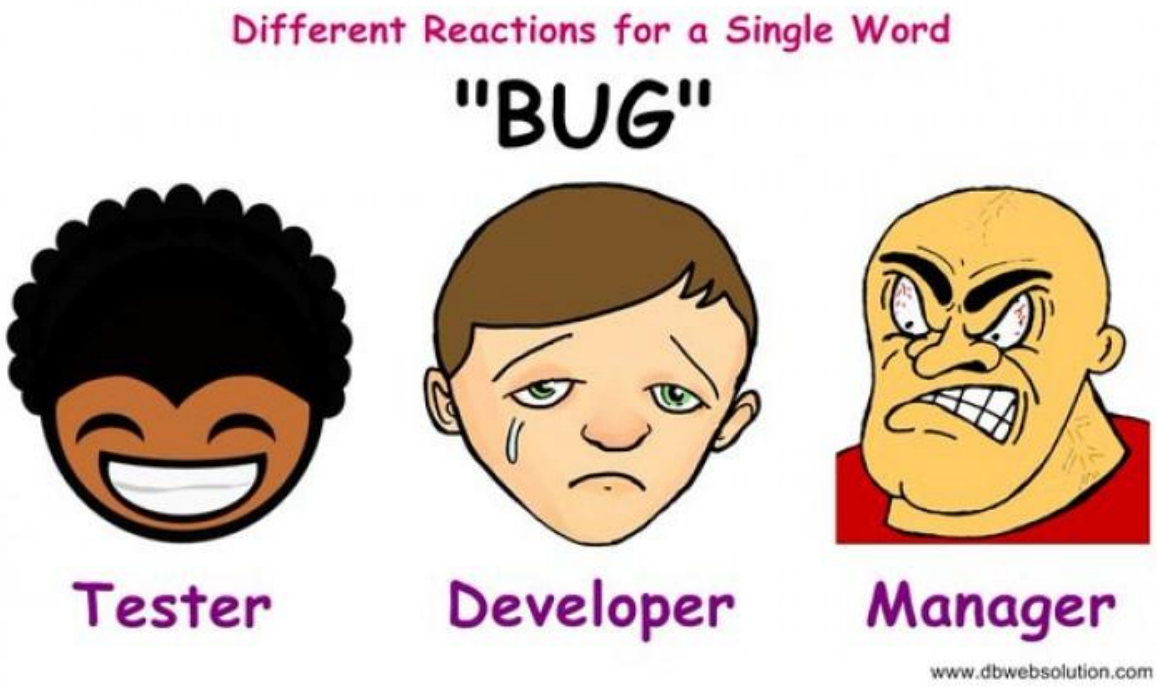
```
double a = 0;
for (int i = 0; i < 10; i++)
    a += 0.1;

double b = 1;

System.out.println("\nComparison accepting some deviation:");
if (Math.abs(a - b) <= 0.001)
    System.out.println("a == b");
else
    System.out.println("a != b");
```

Software Development Testing

- Unit Testing:
  - Floating-point Approximation Errors:

```
double x = 1 / 3.;
double y = x + 1 - 1;
System.out.println("x = " + x);
System.out.println("y = " + y);
if (x == y)
    System.out.println("x == y");
else
    System.out.println("x != y");
```

Software Development Testing



Software Development Testing