SWEN 221
Software Development

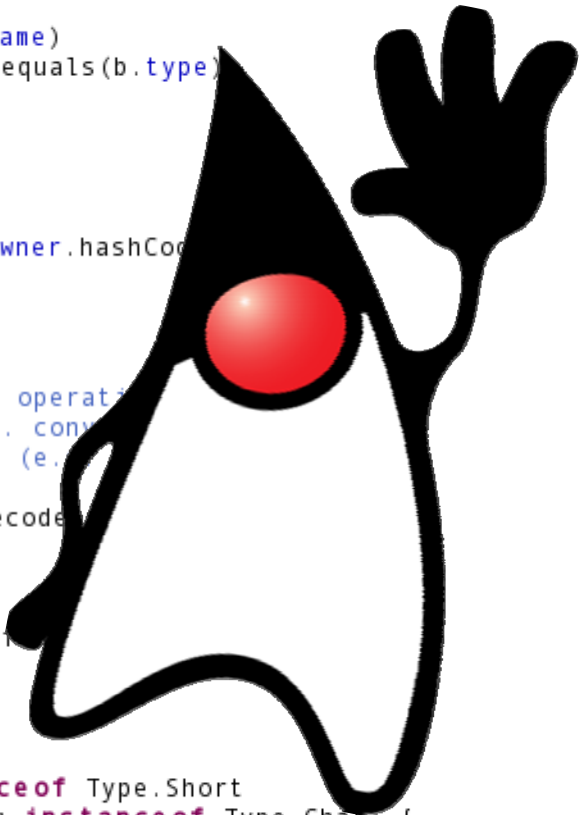
Introduction

Thomas Kuehne & Yi Mei

Victoria University

Welcome!


```
            } else {
                return "invokeinterface " + owner + "." + name + " " + ClassFile.descriptor(type,fals
            }
        }

        public boolean equals(Object o) {
            if (o instanceof Invoke) {
                Invoke b = (Invoke) o;
                return mode == b.mode && name.equals(b.name)
                        && owner.equals(b.owner) && type.equals(b.type)
            }
            return false;
        }

        public int hashCode() {
            return name.hashCode() + type.hashCode() + owner.hashCo
        }
    }

    /**
     * This represents the family of primitive conversion operati
     * i2f, d2f, l2i etc. Observe that in some cases (e.g. con
     * long to a byte) several bytecodes will be produced (e.
     */
    public static final class Conversion extends Bytecode
        public final Type.Primitive from;
        public final Type.Primitive to;

        public Conversion(Type.Primitive from, Type.Prim
            this.from = from;
            this.to = to;

            // Now, sanity check this conversion operator
            if(from instanceof Type.Int || from instanceof Type.Short
                    || from instanceof Type.Byte || from instanceof Type.Char) {
                // i2l, i2f, i2d, i2c, i2b, i2s
                if(to instanceof Type.Long) {
                    return;
                } else if(to instanceof Type.Float) {
                    return;
                } else if(to instanceof Type.Double) {
```



People

- **Alex Potanin** (course coordinator)
  - **Office**: CO 262
  - **Office Hour**: Mon, 10:00—12:00
- **Thomas Kuehne** (lecturer)
  - **Office**: CO 223
  - **Office Hour**: Tue, 16:00—17:00
- **Yi Mei** (lecturer)
  - **Office**: CO 353
  - **Office Hour**: Thu, 11:00—12:00

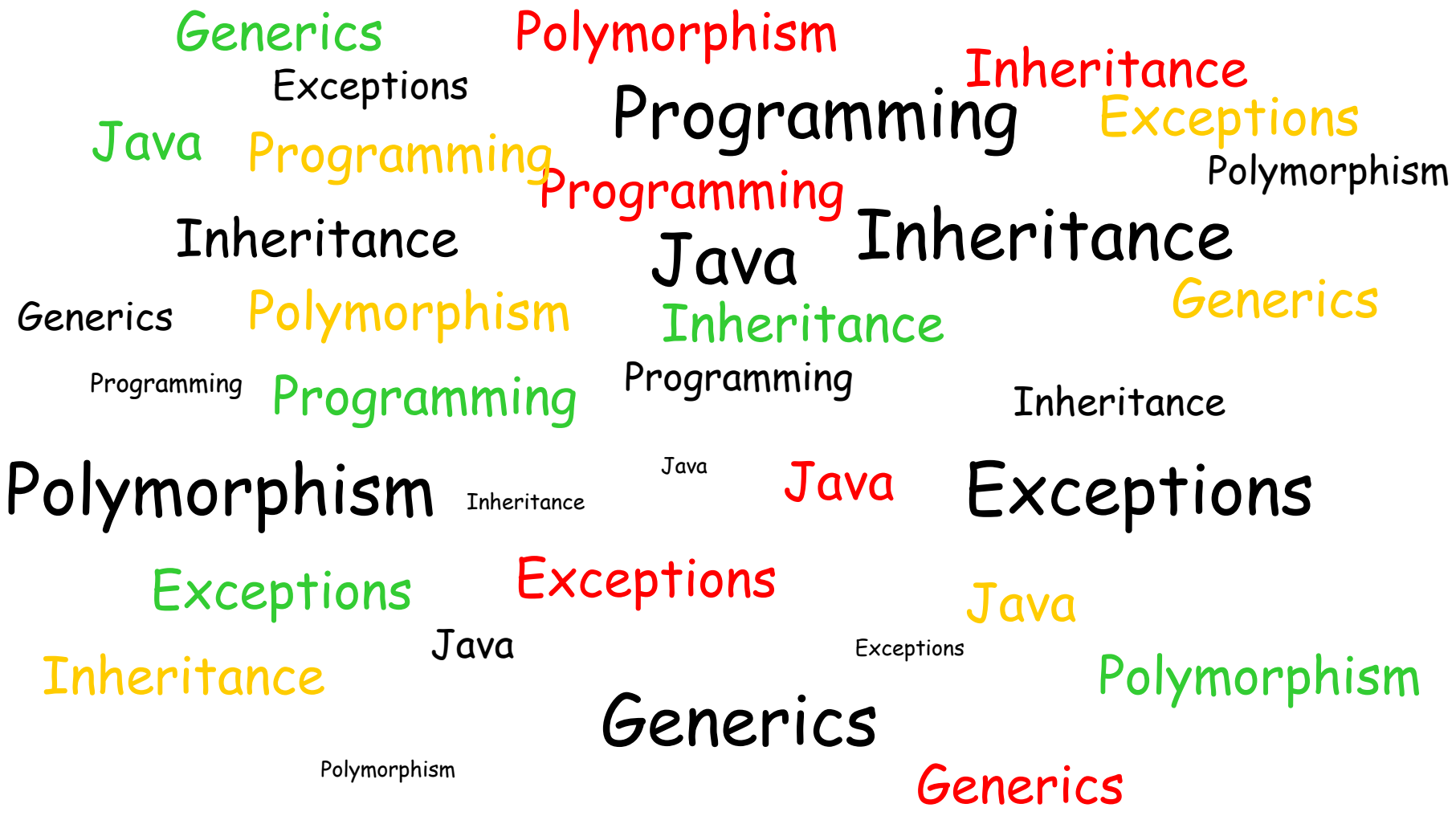
Class Representative

- Opportunity: Become a Class Rep!
- Class Reps are expected to work with the lecturer and the class to support and improve students' learning experiences in your course and at Victoria.
- You will be trained, prepared and supported in your role.
- Representing your class goes towards the VicPlus Award and can lead to other representation opportunities.

TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

VUWSA
YOUR STUDENTS' ASSOCIATION

What is this course about?



Course Announcements

- Lectures – come along!
- Watch out for
  - course-related emails
  - forum posts
- Course website
  - http://www.ecs.vuw.ac.nz/Courses/SWEN221_2017T1

Textbook

There is no official course textbook. However, you should find the following textbook useful

- *Java Foundations: Introduction to Program Design and Data Structures*, by Lewis, DePasquale, and Chase.

Other useful literature include the following:

- *Program Development in Java*, Barbara Liskov
- *Object-Oriented Design & Patterns*, Cay Horstmann, 2nd Ed.
- *Practical Object-Oriented Design*, Bhuvan Unhelkar
- *Effective Java*, Josh Bloch

There are a number of other useful books on programming in Java available in the library!

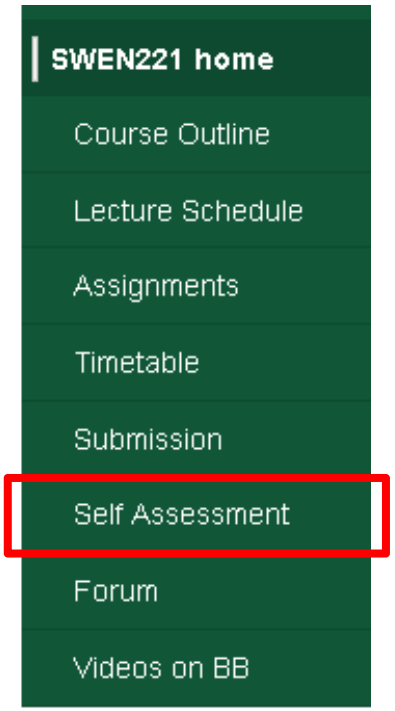
Lectures + Labs

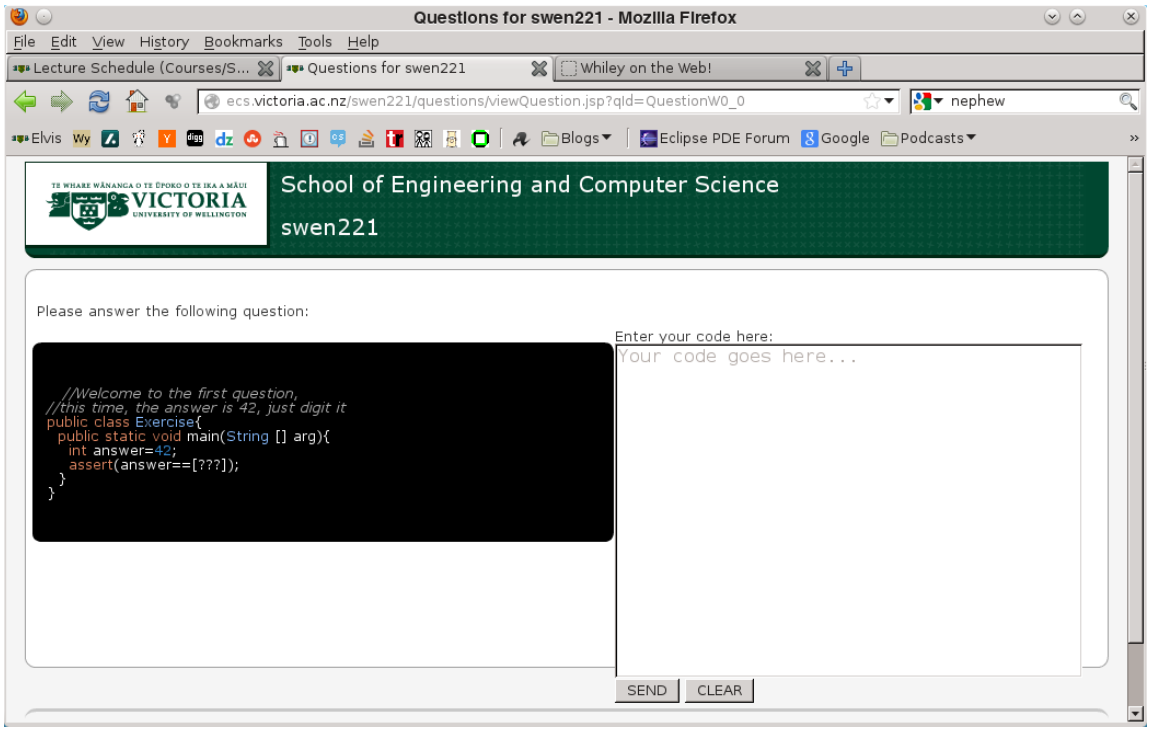
- Lectures
  - **Tuesdays (HM LT205)** and Thursdays **(HM LT205)**, 14:10—15:00
  - **Fridays** (**HM LT205**) will be occasionally used, 14:10—15:00
- Labs
  - **http://ecs.victoria.ac.nz/Courses/SWEN221_2017T1/Timetable**
  - **Rooms:** CO242 and CO243
  - Need to sign up for a lab stream this week!
- Lab Marking
  - Labs will be marked out of 10
  - **Automated marking system** → read the submission notes carefully!
  - **Late Penalty** – **No marks past the Friday deadline**
  - Total worth: **15%**

Assignments

- Five Assignments
  - For most assignments, you'll have two weeks to work on.
  - Together, the assignments are worth **20%**
- Automatic Marking Script
  - Generates **some or all of your marks**
  - Emails you with submission problems, score and failing inputs
  - **You must follow the assignment specification regarding output + submission carefully**
- Marking Criteria
  - **Correctness** – does the code adhere to the given specification?
  - **Style** – code follows style guide, has appropriate comments (inc. Javadoc)
  - **Late penalty** – Up to 24 hours after the deadline → -20%
    between 24 and 48 hours → -40%
    48 hours or more after the deadline → **0%**
    - three days over the entire course are free (no applications necessary) [usable in "1 day" packets only]

Self-Assessment





- Online **Self-Assessment** Tool
  - Currently lists **~100** Java-related questions
  - Answers to questions **due on Thursdays** (23:59)
  - To get the mark, need to complete question **before due date**
  - Overall, the self-assessment contributes 5% to your overall grade
  - **You are encouraged to search the web for answers!**

Self-Assessment Tool

“You should engage in a self-managed learning process. To answer some questions you will need more than what we teach in class. You may have to do some personal research. You are encouraged to seek for answers in cooperation with your class mates.

You can try to answer a question as many times as you like, without losing any mark. This tool wants to encourage you to try, think, try again, talk with your peers, think, learn, try, get it, and then teach others about your experience.”

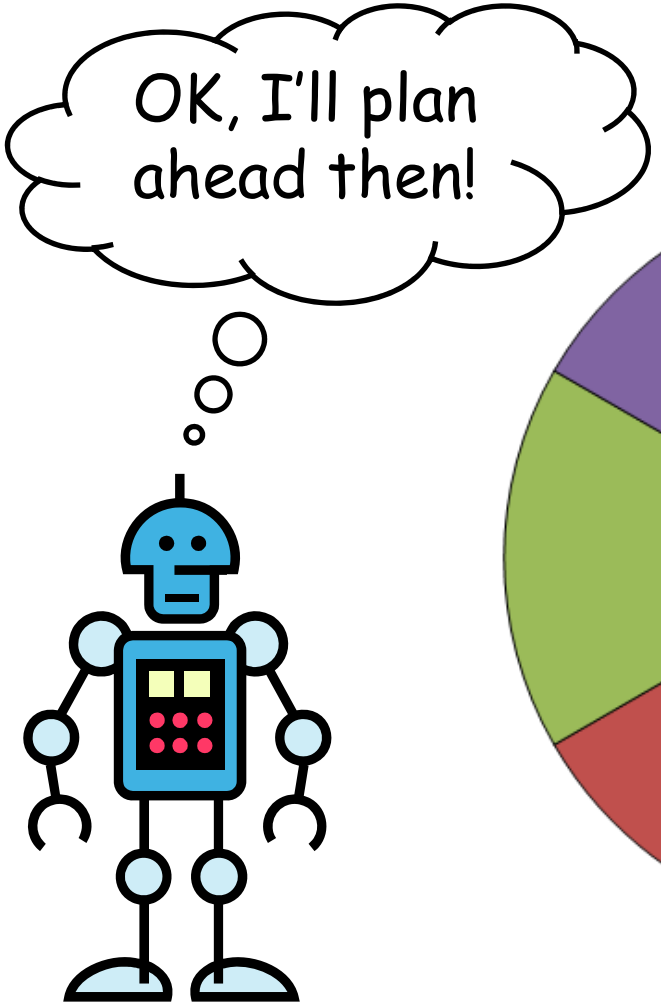
Question status can be

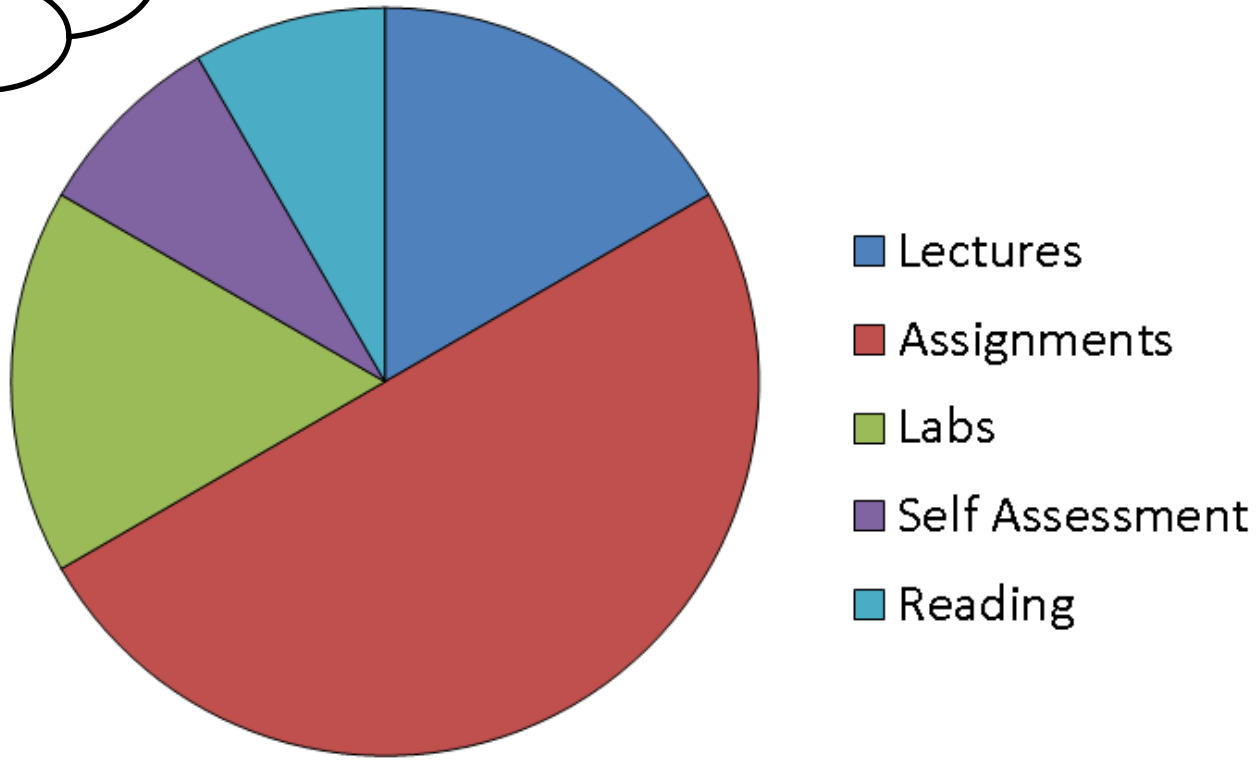
Green,	if you answer correctly and in time
Orange	if you answer correctly but after the deadline
Red	for no answer yet and a passed deadline

You can turn Red questions into Orange ones by answering them.

You should at least get 75 questions right by the end of the course.

Workload





The 10 hours per week workload is a **rough** indicator only!

Assessment

- Assignments 1-5 = **20%**
- Labs = **15%**
- Self-Assessment = **5%**
- Final Exam = **60%**

- **Mandatory Requirements**
  - exam result ≥ 40% (**D**)

- **Passing SWEN 221**
  - meet mandatory requirement(s)
  - overall grade ≥ 50% (**C-**)

Expectations

What's reasonable to assume?

- *"If I attend the lectures, try most of the labs and assignments, then I'll pass."*
- *"I might have to do some extra reading on the internet if I get stuck, or ask a tutor/lecturer for help."*
- *"The tutors/lecturers know everything and they never make mistakes."*
- *"I can pass the course if I copy the assignments from my friend. He/She's a Java Expert!"*
- *"As soon as I get stuck, I ask my friend for help. He/She's great!"*

Rules And Policies

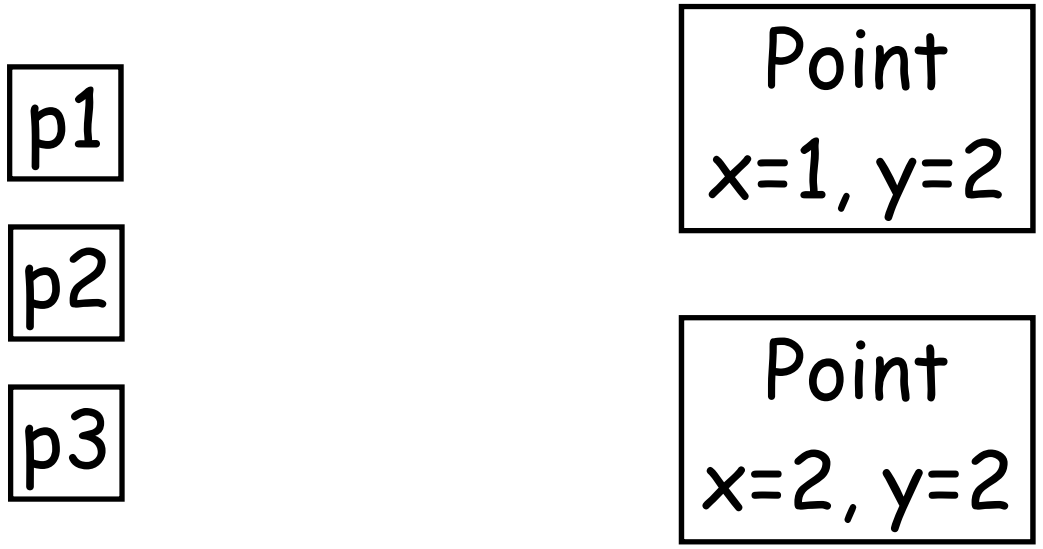
- Standard Policies
  - Academic Integrity / Plagiarism
    - *submit your own work only or add a declaration*
  - Student Support
  - Student & Staff Conduct

Equality Quiz – what gets printed?

```
class Point {
 int x; int y;
 public Point(int x, int y) {
  this.x = x; this.y = y;
} }

Point p1 = new Point(1, 2);
Point p2 = p1;
Point p3 = new Point(2, 2);
p2.x = 2;
if (p1 == p2) { System.out.println("p1==p2!"); }
if (p1 == p3) { System.out.println("p1==p3!"); }
```

A) "p1==p2!" "p1==p3!" B) Nothing C) "p1==p2!"

Why?



- The '==' operator
  - Checks whether two references point to the same *object*
  - does not not compare *values*
    - Must override Object:equals() for value comparisons

Quiz: What gets printed?

```
class Point {
 int x = 0;
 int y = 0;
 static int z = 0;
 Point() { z++; }
}

Point p1 = new Point();
Point p2 = new Point();
System.out.println("x = " + p2.x);
System.out.println("y = " + p2.y);
System.out.println("z = " + p2.z);
```

A) "x = 0"
"y = 0"
"z = 0"

B) "x = 1"
"y = 0"
"z = 1"

C) "x = 0"
"y = 0"
"z = 2"