Static methods & variables

• Static method: method that is not invoked on an object. Static variables: exists even if no objects exist

• In Java, these are called class methods and class variables because they have no objects associated with them.

Purpose: Static Method

• Provide a function / service that uses only variables that are passed into it
  – Java requires ALL functions (methods) to be associated with a class

• sqrt
• ceiling, floor
• exponent

Purpose: Static Method

```java
public class MathFunctions {
  public static PI = 3.1415blahblahblah;
  public static float circle_circumference(float radius)
  { return 2*PI*radius;}
  public static float circle_area(float radius)
  { return PI * radius * radius; }
}
```
Logistics: Static Method

- May not use any instance variables
- May not call any instance methods
- Can only access input parameters, local variables, class variables
- Just like the C functions you’re used to!

Purpose: Static Variable

- Provide a constant (e.g. PI) that you can use without setting it
- Provide a variable that aggregates information across all objects of a particular class (be careful – easy to get runtime race errors)

```java
public class BankAccount {
    private double balance;
    private int accountNumber;
    private static int lastAssignedNumber = 1000;

    public BankAccount() {
        balance = 0;
        // These two instructions need to complete without interruption.
        lastAssignedNumber++;
        accountNumber = lastAssignedNumber;
    }

    // ....
}
```

Logistics: Static Variable

- Only one exists per class – every object of that class shares that one variable
What happens if I “sprinkle” static until it compiles?

• If everything (methods + variables), then what happens?

— Compiles file
— Works if you have only one object of that type
— Fails if you expected to have separate copies of variables for each object when you make multiple instances of the object.

Now we know all parts...

• Public static void main
  — public:
  — static:
  — void:

What happens if I “sprinkle” static until it compiles?

• If everything (methods + variables), then what happens?

  — Compiles file
  — Works if you have only one object of that type
  — Fails if you expected to have separate copies of variables for each object when you make multiple instances of the object.

Now we know all parts...

• Public static void main
  — public: allows anything to call it
  — static: This solve the chicken & egg problem – you don’t need an object before program starts
  — void: doesn’t return anything