

# Tutorial Sheet - Topic 03

## Programming Fundamentals 1

Data Types, Variables & Expressions

### Tutorial Questions - Variables, Data Types & Operators

#### 1. Concept Check

- What is a variable in Java, and why do we need to specify its data type?
- Give two examples of valid variable names and two invalid ones. Explain why.

#### 2. Keywords

- What is a *keyword* (reserved word) in Java?
- Why can't you name a variable `class`?

#### 3. Tokens

Break the line below into its tokens and identify their roles:

```
double price = 19.99;
```

#### 4. Primitive Data Types

- What is the difference between `int` and `double`?
- When would you choose a `boolean` instead of an `int`?

#### 5. Errors

- Why is `Int x = 5;` invalid?
- What happens if you try `int x = 5.5;`?

#### 6. Operators

- a. What is the difference between `/` when used with integers vs. doubles? b. Predict the output of:

```
int a = 10, b = 6;
float c = 6f;
System.out.println(a / b);
System.out.println(a / c);
```

## 7. Discussion

---

- a. Why might `gearRatio` be a better variable name than `gr`?
  - b. How does using clear variable names make it easier to read a program and fix any logical errors?
- 

# Tutorial Questions - Local Variables & Scope

---

## 8. Local Variables

---

- a. How are local variables different from fields?
- b. When are local variables created and destroyed?

## 9. Scope

---

- a. What do we mean by the scope of a variable?
- b. Explain why the following will not compile:

```
{
    int x = 5;
}
System.out.println(x);
```

## 10. Refund Example

---

- a. If we wish the `refundBalance` method to reset the balance to zero and return the original balance, why does this method fail?

```
public int refundBalance() {
    return balance;
    balance = 0;
}
```

- b. How can we fix it?

## 11. Block Scope

---

In the method below, which lines will compile successfully?

```
public void scopeDemo(int param) {
    int localVar = 5;
    {
        int innerVar = 10;
        System.out.println(param + localVar + innerVar); // Line A
    }
    System.out.println(param + localVar); // Line B
    System.out.println(innerVar); // Line C
}
```

## 12. Lifetime

---

What is the lifetime of:

- a. a field?
- b. a local variable?

## 13. Thinking Question

---

- a. Why does Java restrict access to variables outside their scope?
- b. How does this help prevent bugs?