Conclusion of Lecture 3: Java Basics 2

1. **Operators:**
    - **Arithmetic Operators (`+, -, *, /, %`):** Perform mathematical calculations. Integer division truncates the decimal part. The modulus operator `%` returns the remainder and is useful for checking even/odd numbers.
    - **Operator Precedence:** Determines the order of evaluation in an expression (e.g., `*`, `/`, `%` are evaluated before `+`, `-`).
    - **Increment/Decrement Operators (`++, --`):**
        - **Prefix (e.g., `++var`):** Increments/decrements the variable first, then uses the new value.
        - **Postfix (e.g., `var++`):** Uses the current value first, then increments/decrements the variable.
    - **Relational Operators (`==, !=, >, <, >=, <=`):** Compare two values and return a boolean result (`true` or `false`).
    - **String Comparison:** Use the `.equals()` method to compare the *contents* of Strings, not the `==` operator.
    - **Logical Operators (`&&, ||, !`):** Combine boolean expressions using AND, OR, and NOT logic, as defined by their truth tables.
    - **Compound Assignment Operators (`+=, -=, *=, /=, %=`):** Shorthand for operations like `x = x + 5` which becomes `x += 5`. They perform an implicit cast.
2. **Scanner Class:**
    - Used to get input from the user via the keyboard.
    - Must be imported: `import java.util.Scanner;`

- Must be created: `Scanner input = new Scanner(System.in);`
- Uses specific methods to read data: `nextInt()`, `nextDouble()`, `next().charAt(0)`, `next()`, etc.

3. **Error Types:**
   - **Syntax Errors:** Compile-time errors caused by violating Java's language rules (e.g., missing semicolon, mismatched braces). The compiler catches these.
   - **Runtime Errors:** Errors that cause the program to crash during execution (e.g., division by zero).
   - **Logic Errors:** The program runs but produces incorrect results due to a flaw in the algorithm or implementation (e.g., using `* 2` instead of `* 3` to calculate a triple).
4. **Debugging:** The process of identifying, analyzing, and fixing errors in code.

Practice Problems & Questions

Part 1: Arithmetic, Relational, and Logical Operators

Q1: What is the result of `10 / 4` in Java? Why?
A1: The result is `2`. Because both `10` and `4` are integers, Java performs integer division, which truncates the decimal part.

Q2: What is the result of `10.0 / 4`?
A2: The result is `2.5`. Because `10.0` is a `double`, Java promotes the calculation to floating-point division.

Q3: What is the value of `result`?

```
int x = 17;
int result = x % 4;
```

A3: `1`. The remainder of 17 divided by 4 is 1.

Q4: Evaluate the following expression, showing the order of operations:
`3 + 4 * 5 - 6 / 2`
A4:

1. `4 * 5 = 20`
2. `6 / 2 = 3`
3. `3 + 20 = 23`
4. `23 - 3 = 20`
   The result is `20`.

Q5: What is the value of the boolean variable `isPositive`?

```
int number = -5;
boolean isPositive = (number > 0);
```

A5: `false`

Q6: Given `int a = 5, b = 10;`, what is the value of `(a < 5) && (b > 10)`?
A6: `false`. The AND (`&&`) operator requires both sides to be true. `(5 < 5)` is false, so the entire expression is false.

Q7: Given `int a = 5, b = 10;`, what is the value of `(a <= 5) || (b > 10)`?
A7: `true`. The OR (`||`) operator requires only one side to be true. `(5 <= 5)` is true, so the entire expression is true.

Q8: Why is `str1 == str2` an incorrect way to compare the Strings below? How should it be done?

```
String str1 = "hello";
String str2 = "hell" + "o";
```

A8: `==` compares object references, not content. While it might work in some cases due to Java's string pool, it is unreliable. The correct way is to use `str1.equals(str2)`.

Part 2: Increment/Decrement Operators

Q9: What are the final values of `a` and `b`?

```
int a = 5;
int b = a++;
```

A9: `a` becomes `6`, `b` is assigned `5` (the original value of `a` before the increment).

Q10: What are the final values of `a` and `b`?

```
int a = 5;
int b = ++a;
```

A10: `a` becomes `6`, `b` is assigned `6` (the new value of `a` after the increment).

Q11: What is the output of the following code?

```
int count = 10;
System.out.println(count++);
System.out.println(++count);
System.out.println(--count);
System.out.println(count--);
```

A11:

```
10
12
11
11
```

Part 3: Compound Assignment & Scanner

Q12: Rewrite this code using a compound assignment operator: `sum = sum + 15;`

A12: `sum += 15;`

Q13: What is the value of `num` after this statement executes?

```
int num = 10;
num *= 2 + 3; // num = num * (2 + 3)
```

A13: `50`. The right-hand side is evaluated first: `2 + 3 = 5`, then `num * 5 = 50`.

Q14: Write the Java code to create a Scanner object named `keyboard` and use it to read an integer from the user into a variable named `userAge`.
A14:

```
import java.util.Scanner; // This would be at the top of the file

Scanner keyboard = new Scanner(System.in);
System.out.print("Enter your age: ");
int userAge = keyboard.nextInt();
```

Part 4: Error Identification

Q15: Identify the type of error in each example:
a) `System.out.println("Hello World")` // Missing semicolon
b) `System.out.println(1 / 0);` // Division by zero at runtime
c) `System.out.println("2 + 2 = " + (2 + 2));` // This is actually correct. A logic error would be `"2 + 2 = " + 2 * 2` which prints "2 + 2 = 4", misleading the user.

A15:
a) **Syntax Error**
b) **Runtime Error**
c) *(Trick Question)* This is correct. A true logic error would be `System.out.println("2 + 2 = " + 2 * 2);` which gives the right answer for the wrong reason, or `System.out.println("2 + 2 = 5");` which is just wrong.

Q16: Find and fix the 5 errors in the code from the lecture:

```
/* This is an exercise
public IncorrectProgram
```

```
{
    public void Main (String args)
    {
    System.out.println ("I study Java " + "one")
    System.out.println ("3 + 7 = + (7 * 3));
}
```

A16:

1. **Missing closing comment delimiter `*/`.**
2. **Class header missing `class` keyword.** Should be `public class IncorrectProgram`.
3. **Method `main` is misspelled as `Main` and has wrong signature.** Should be `public static void main(String[] args)`.
4. **Missing semicolon on the first `println` statement.**
5. **Mismatched quotes and parentheses in the second `println` statement.** Should be `System.out.println("3 + 7 = " + (7 + 3));` (assuming the intent was to add).

Part 5: Programming Challenges

Q17: Write a program that asks the user for a temperature in Fahrenheit and converts it to Celsius. The formula is: `celsius = (5.0/9.0) * (fahrenheit - 32)`.

A17:

```
import java.util.Scanner;

public class FahrenheitToCelsius {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter temperature in Fahrenheit: ");
        double fahrenheit = input.nextDouble();

        double celsius = (5.0 / 9.0) * (fahrenheit - 32);
```

```
        System.out.println(fahrenheit + "F is equal to " + celsius +
"C");
    }
}
```

Q18: Write a program that reads an integer from the user and prints whether it is even or odd. (Hint: Use the modulus operator `%`).
A18:

```
import java.util.Scanner;

public class EvenOrOdd {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        int number = input.nextInt();

        if (number % 2 == 0) {
            System.out.println(number + " is even.");
        } else {
            System.out.println(number + " is odd.");
        }
    }
}
```