

Theory Class

Java Basics

Muhammad Jamil

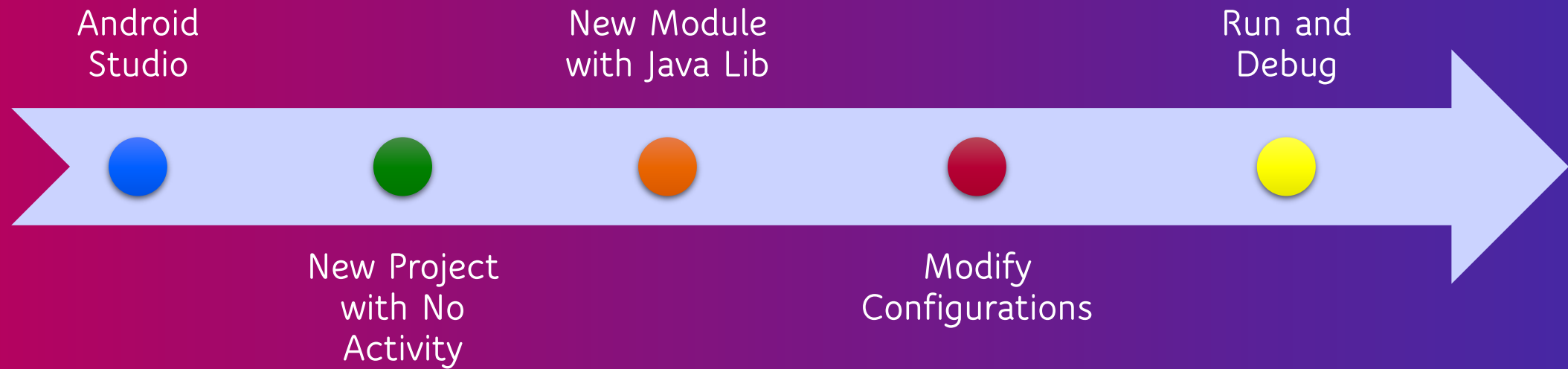
Software Developer

Agenda

Today's Discussion

- Writing Java programs
- Edit-Compile-Execute
- Bugs, bugs bugs!
- Types of program errors
- Practical exercises

Run Pure Java on Android Studio



Hands On Java

- Try Java yourself
- There's nothing like practice when learning a new programming language!
- Just play around, don't be put off when something doesn't work –it happens to all of us!
- And it's more fun, too!

Tasks

- Print your First Name, Last Name, Email, Contact Number and Address on Different lines
- Solve this equation using mathematical rules and print the value of unknown x

$$x = 55 + 44 - 30 * 2$$

- The radius of the circle is 3.2 cm. The formula to calculate the area of a circle is $2\pi r$. You need to find the area using this formula.

(Pi = 3.1416)

- Convert 5.68 miles into kilometers while 1 mile = 1.60934

Comments

Two types of **comments** in Java:

Line comments

Block comments

Comments are ignored by the compiler. You can write whatever you want!

Comments are good practice. Make your code understandable to others and also to yourself

So please comment your code!

Line Comments

- `//` indicates the beginning of a comment
- All text between these symbols and the end of the line are ignored.
- Place `//` at each line if you want multiline comments (or use block comments)

Example:

```
int x = 1 + 1 - 2; // x is 0
```

Block Comments

- Anything between the symbol pairs `/*` and `*/` is considered a comment
- This can span multiple lines
- This is used to explain what a whole block, a method or a class is for
- It can also hold other information like author etc.
- Variant: `/** ... */` (used by javadoc)

Bugs

- A mistake in a program is called a **bug**
- Like real bugs they usually hide and are not easily detected
- We neither want bugs at home nor in our programs
- The process of eliminating mistakes (bugs) in our program is called **debugging**

Many IDEs offer a useful tool called a **debugger**

pes of bugs (or **errors**):

- *Syntax error*
- *Run-time error*
- *Logic error*



Syntax and Semantics

Syntax

- Legal arrangement of words and punctuation
- Grammar rules of a language
- Symbols, words, etc.

Semantics

- **Meaning** of a program (adhering to the given syntax)
- What happens when you run the program

Syntax Error

- Grammatical mistake in your program, for instance Misspelled keyword (e.g., `calss`)
- Missing punctuation
- Misspelled (or unknown) variable
- Compiler catches syntax errors
- You can't argue with the compiler!

Run-time Error

- Not detected by the compiler, only when executing the code
- Very common in Java: the notorious `NullPointerException`
- Run-time errors may also be caused by failing communication links, insufficient memory, etc.

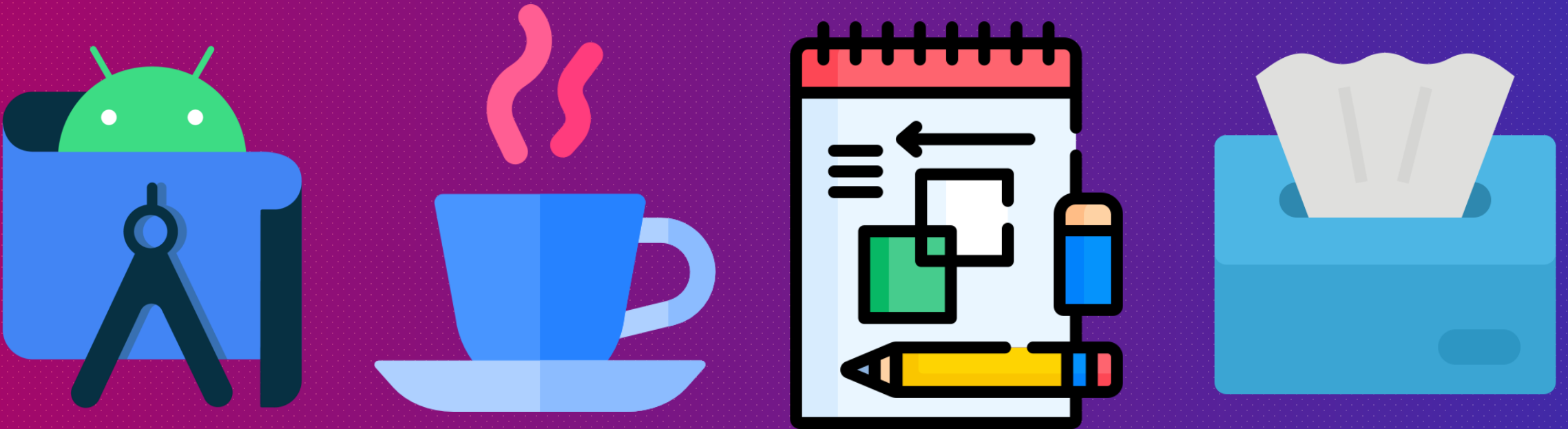
Logic Error

- Mistake in the underlying algorithm
- Not detected by the compiler -there may even not be an error message!
- Program doesn't do what you want it to do
- your intended semantics are different from the program's actual ones
- Example: if you want to print $1+1$ but write
`System.out.println(1 -1);`

Test your code

- Debugging and testing is an essential part in the software development process!
- Especially test your code with respect to logic errors
- Learn to read and understand error messages

Soft Reminder Again





THANK YOU!

Theory Questions?