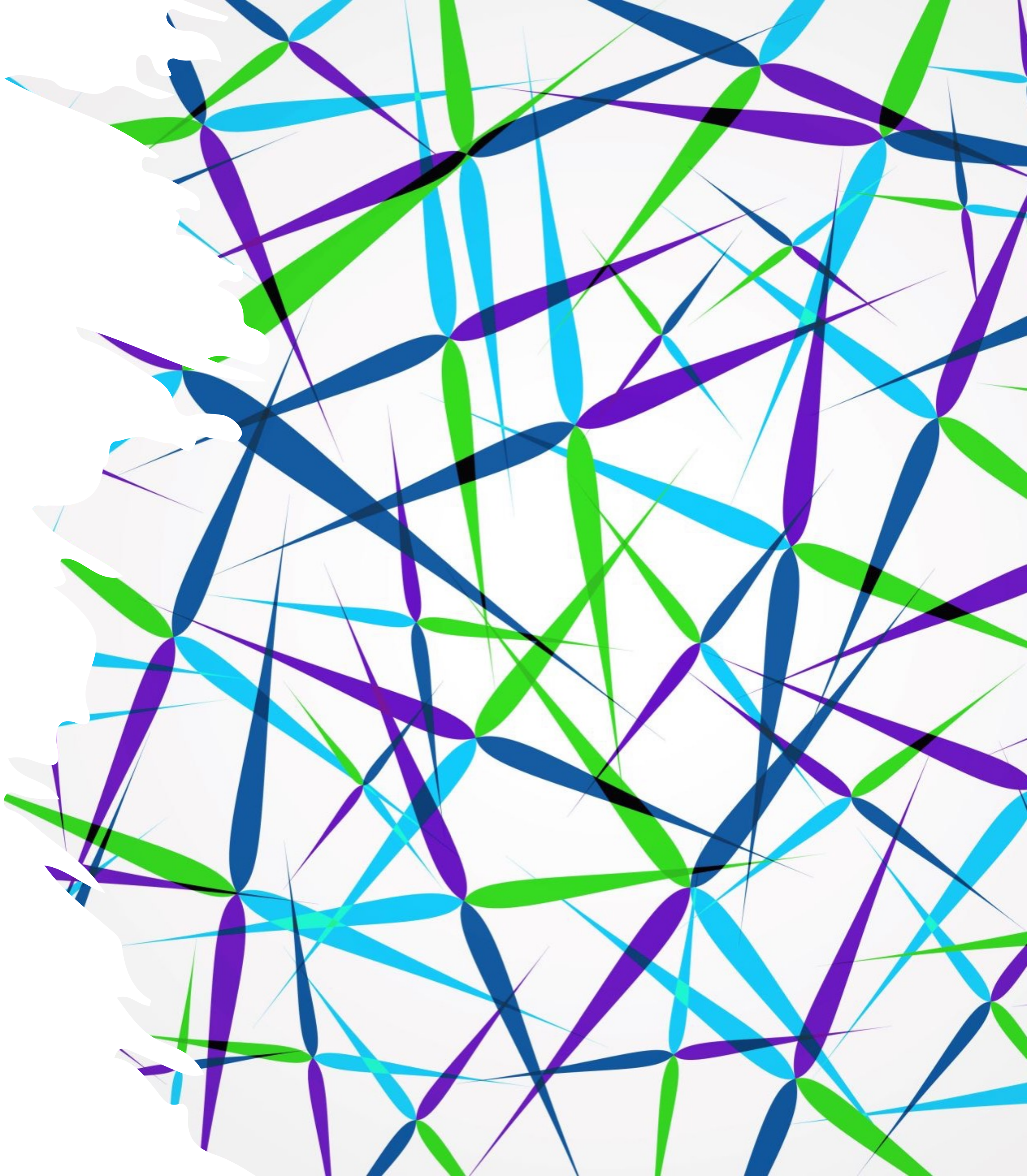
L1_OOPD_Java

PUSHPENDRA SINGH



OOPD Course Introduction

- Object-oriented Programming and Design (OOPD)
- 2 Credits
- One class every week for 13 weeks
    - CSE: Tuesday
    - ECE+CB: Friday

- Instructor
    - Pushpendra Singh
        - A-502, R&D Building
    - Office hours:
        - Tue/Fri: 04:30 – 5:30 p.m.

OOPD Course

- *We are learning OOPD and not Java or Python*
  - I will use Java Examples; students can do assignments, etc. in Java or Python

- Methodology
  - Learning by doing
  - Programming will be the mainstream and classes will only anchor the learning
  - Bring laptop to class everyday

- Google Classroom & Slack
  - TAs are already added

OOPD Evaluation

- Mid-sem
  - 10 Marks
- End-sem
  - 15 Marks
- Programming Assignments & Quizzes
  - 75 Marks
- Grades would be relative

Java: 1996 and beyond (Java 17)

- Simple
- Object-Oriented
- Distributed
- Robust
- Secure
- Architecture-Neutral
- Portable
- Interpreted
- High-Performance
- Multithreaded
- Dynamic

Java/Python Development Environment - DIY

Choose

Choose any IDE that you want

- Eclipse, IntellyJ, VS Code…

Install

Install JDK/Python and set-up the IDE for Java/Python programming

Java Basics

```
/**
 * This is the first sample program in Core Java Chapter 3
 * @version 1.01 1997-03-22
 * @author Gary Cornell
 */
public class FirstSample
{
   public static void main(String[] args)
   {
      System.out.println("We will not use 'Hello, World!'");
   }
}
```

Data Types

- Java is a strongly typed language.
- Every variable must have a declared type.
- 8 types
  - 4 integers
  - 2 float
  - 1 char
  - 1 boolean

Integer types

- Integer Types
  - int
  - short
  - long
  - Byte
- Float types
  - float
  - double

- Character Type
  - char
- Boolean Type
  - boolean

Variables & Constants

- All variables must be declared before use
- All variables must be explicitly initialized before use.

```
double salary = 65000.0;
System.out.println(salary);
int vacationDays = 12; // OK to declare a variable here
final double CM_TO_INCH = 2.54; //constant
```

Question

- What will we have to do if we want to declare a constant for multiple methods in a classes in the program?

Enumerated Types

```
enum Size { SMALL, MEDIUM, LARGE, EXTRA_LARGE };
```

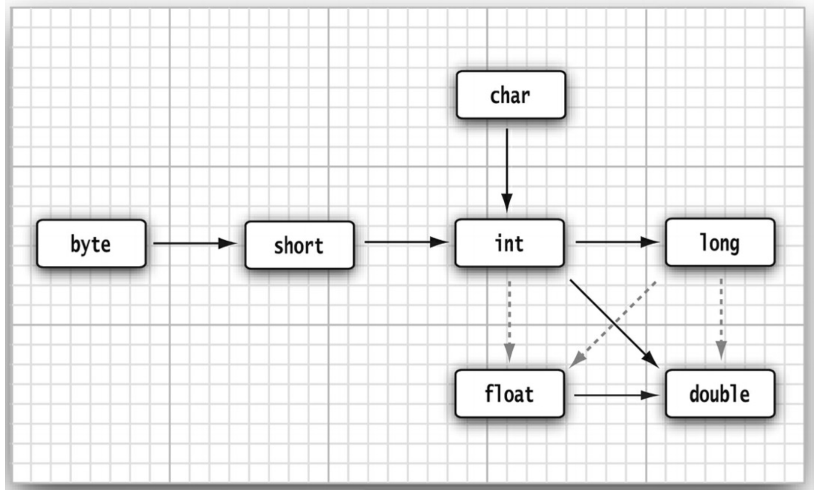
```
Size s = Size.MEDIUM;
```

- Enumerated types are used to limit the values a variable can take.

Conversion between types

Conversions in which loss of information is possible are done by means of casts.

```
double x = 9.997;
int nx = (int) x;
```



Strings

- Java strings are sequences of Unicode characters.

- String e = ""; // an empty string
- String greeting1 = "Hello";
- String greeting2 = ”World";
- System.out.println(greeting 1 + greeting 2);
- System.out.println(greeting1.substring(o,3);

Strings

- How would you change the greeting1 to “Help”?
- How many Strings object are present?

Reading Input

- Standard input stream: System.in

```
Scanner in = new Scanner(System.in);
System.out.print("What is your name? ");
String name = in.nextLine();
```

Formatting Output

```
double x = 10000.0 / 3.0;
System.out.print(x);
```

prints 3333.3333333333335

```
System.out.printf("%8.2f", x);
3333.33
```

File Input Output

```
Scanner in = new Scanner(Path.of("myfile.txt"),
StandardCharsets.UTF_8);
```

```
PrintWriter out = new PrintWriter("myfile.txt",
StandardCharsets.UTF_8);
```

Control Flow

```
public static void main(String[] args)
{
   int n;
   . . .
   {
     int k;
     . . .
   } // k is only defined up to here
}
```

Control Flow

- Conditional Statements
  - If-else
  - switch
    - Remember to break after each case
- Loops
  - while or do-while
  - for
- Breaking control flow
  - break; labled break; continue

Array, For Each, and rest

- Array
  - Single dimension
  - Multi-dimension
- For each
  - for(element:collection){…}
- BigInteger and BigDecimal