International study centre

AN INTRODUCTION TO OBJECT-ORIENTATION AND THE JAVA PROGRAMMING LANGUAGE

- Name : John Alamina
- Email : john.alamina@hud.ac.uk

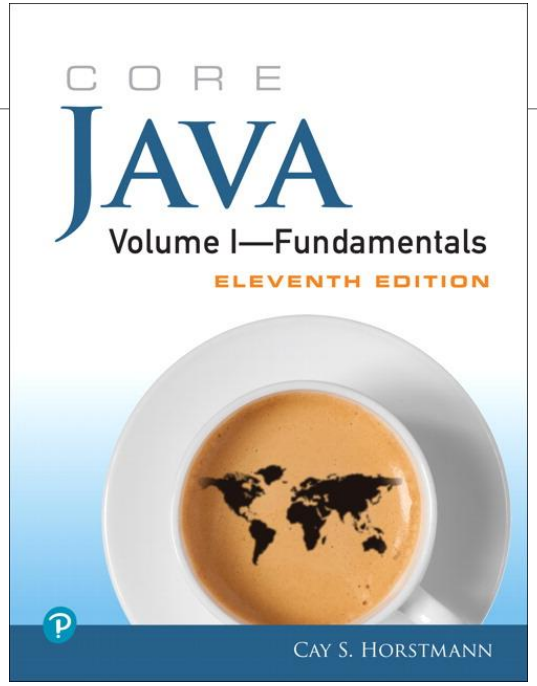
Java Bootcamp Outline

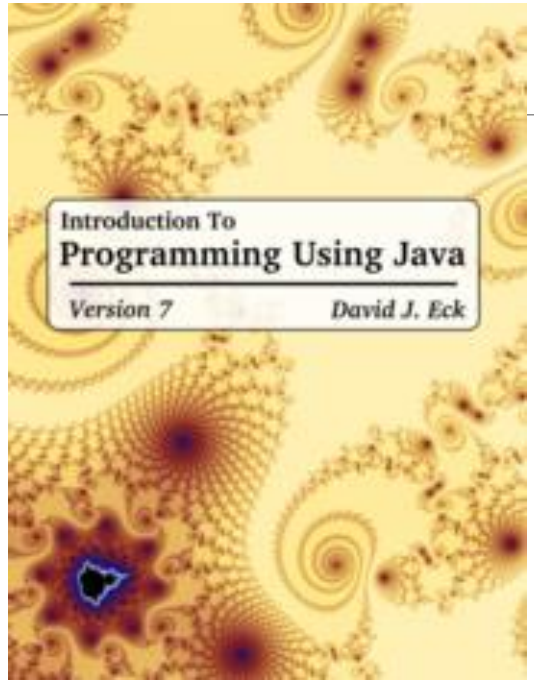
- Lesson 1 – Java Vs C++
- Lesson 2 – Structured Programming in Java
- Lesson 3 – Introduction to Object-oriented programming in Java
- Lesson 4 – More OOP and Core Java language topics
- Lesson 5 – Java Generics and Java Collections
- Lesson 6 – GUI development and Exception Handling

Ground rules

1. Attend all the lessons.
2. Do any pre-class exercise in preparation for the class
3. 5 minute rule and synchronous learning etiquettes
4. Ensure your camera is on at all times
5. Use a laptop for all sessions
6. Participate in all the polls within 2 minutes
7. Stay till the end of the class to do your exercises
8. Contact your tutors (john.alamina@hud.ac.uk, onyedikachi.ulelu@hud.ac.uk)

Supplementary material





- The Java Tutorial
- Java API documentation
- Link to today's Session Screencast
- Link to John's Group Padlet
- Link to Kelly's Group Padlet

Lesson 1 - Outline

- Classes and objects in a nutshell
- The hello Java class.
- Java Variables and Datatypes.
- Comparing Java to C++
- Operations in Java
- Functions in Java (they are now called methods)

Classes and objects in a nutshell

- Can you run the bicycle class on your computer?

```
class Bicycle {

    int cadence = 0;
    int speed = 0;
    int gear = 1;

    void changeCadence(int newValue) {
        cadence = newValue;
    }

    void changeGear(int newValue) {
        gear = newValue;
    }

    void speedUp(int increment) {
        speed = speed + increment;
    }

    void applyBrakes(int decrement) {
        speed = speed - decrement;
    }

    void printStates() {
        System.out.println("cadence:" +
            cadence + " speed:" +
            speed + " gear:" + gear);
    }
}
```

A Class

Another class

```
class BicycleDemo {
    public static void main(String[] args) {

        // Create two different
        // Bicycle objects
        Bicycle bike1 = new Bicycle();
        Bicycle bike2 = new Bicycle();

        // Invoke methods on
        // those objects
        bike1.changeCadence(50);
        bike1.speedUp(10);
        bike1.changeGear(2);
        bike1.printStates();

        bike2.changeCadence(50);
        bike2.speedUp(10);
        bike2.changeGear(2);
        bike2.changeCadence(40);
        bike2.speedUp(10);
        bike2.changeGear(3);
        bike2.printStates();
    }
}
```

2 objects

- https://docs.oracle.com/javase/tutorial/java/concepts/class.html

Sample Bicycle Class

- A Bicycle has (attributes)
  1. Cadence
  2. Speed
  3. Gear
- A Bicycle can (behaviour)
  - Change Cadence
  - Change speed
  - Change gear

```
class BicycleDemo {
   public static void main(String[] args) {

      // Create two different
      // Bicycle objects
      Bicycle bike1 = new Bicycle();
      Bicycle bike2 = new Bicycle();

      // Invoke methods on
      // those objects
      bike1.changeCadence(50);
      bike1.speedUp(10);
      bike1.changeGear(2);
      bike1.printStates();

      bike2.changeCadence(50);
      bike2.speedUp(10);
      bike2.changeGear(2);
      bike2.changeCadence(40);
      bike2.speedUp(10);
      bike2.changeGear(3);
      bike2.printStates();
   }
}
```

- **Classes are factories for creating objects**

Hello Java

```
class Main {
  public static void main(String[] args) {
    System.out.println("Hello world!");
  }
}
```

Hello C++

```
#include <iostream>
using namespace std;
int main()
{
cout << “Hello World";
  return 0;
}
```

Any Questions?

What is a variable?

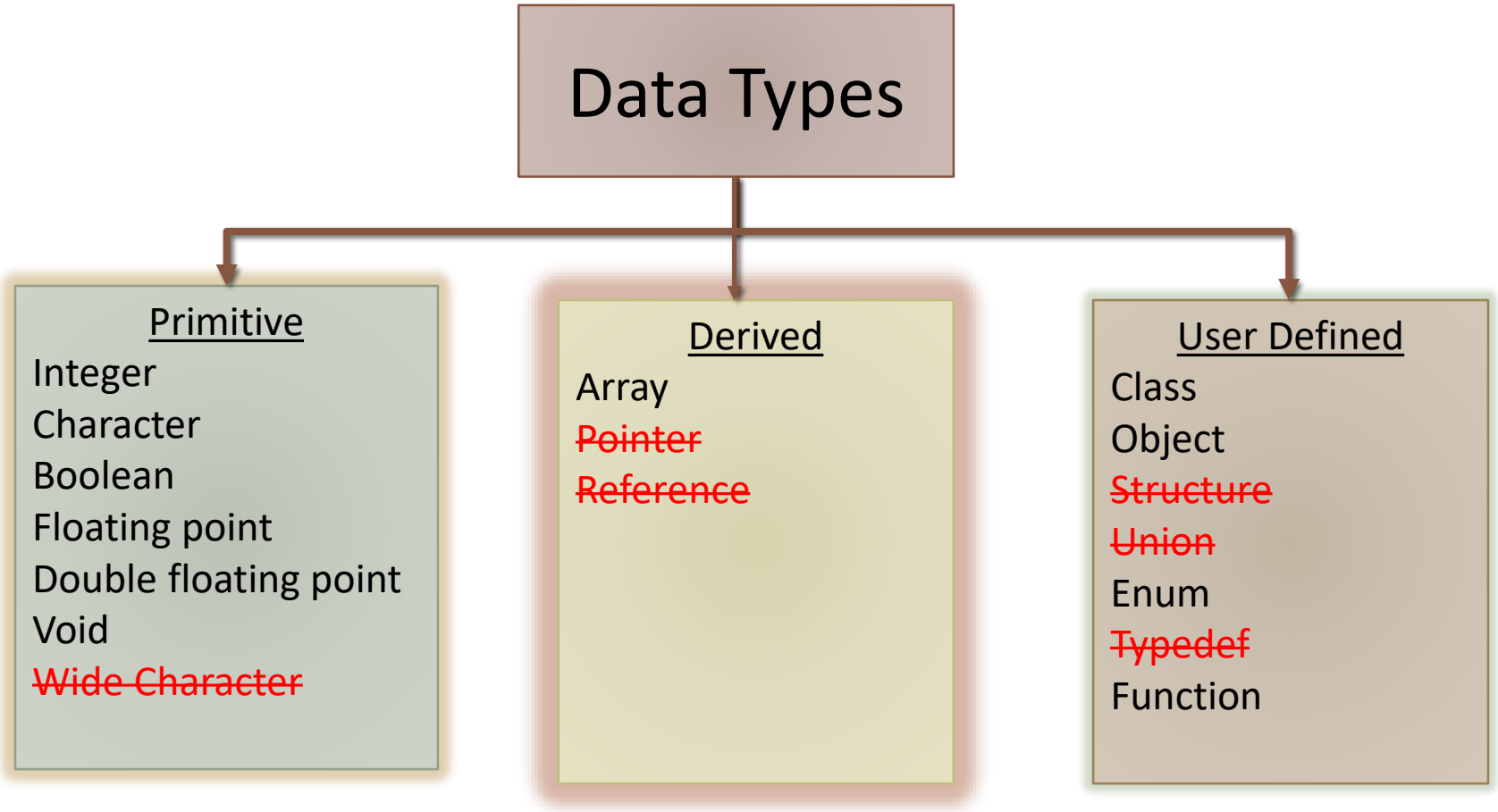
- A name given by the programmer within a program that is not reserved and can store values. Identify the variables in the following program:

```
package com.studygroup;

public class Main {

    public static void main(String[] args) {
        float xx = 5; //int = integer type (whole numbers)
        float y = 7;// = is assignment
        System.out.println((xx + y) + " " + (xx / y));
    }
}
```

Data Types in Object Oriented Programming



Java primitive data types

- byte - 8-bit signed two's complement integer (-128 to +127)
- short - 16-bit signed two's complement integer. (-32768 to +32767)
- int - 32-bit signed two's complement integer (~-2m to ~+2m)
- long – 64 -bit signed two's complement integer
- float - 32-bit IEEE 754 floating point. 7 decimal places accuracy
- double – 64 -bit IEEE 754 floating point. 15 decimal places accuracy
- char - 16-bit Unicode character.
- boolean - This data type represents one bit of information (i.e. true or false)

Any Questions?

Java vs C++

```
package com.studygroup;


public class Main {


    public static void main(String[] args) {
        float x = 5; //int = integer type (whole numbers)
        float y = 7;// = is assignment
        System.out.println((x + y) + " " + (x / y));
    }
}
```

```
#include <iostream>
using namespace std;

int main()
{
float x = 5; //int = integer type (whole numbers)
float y = 7;// = is assignment
cout << "\n";
cout << x + y << " " << x / y;
cout << "\n";
return 0;
}
```

Java vs C++

SIMILARITIES	DIFFERENCES
1. Use of curly braces	1. Code organization
2. Basic syntax	1a. Package vs namespace
2a. Command/sequential syntax	2. Entry Class/method
2b. Control structures	3. Explicit Type modifiers
3. Case sensitivity	4. User-defined operator-overloading
4. Some keywords	5. Some keywords
5. Method syntax	6. Classes and filenames
6. Operators	7. Include vs import
7. Literal values e.g. string quotes	8. Method headers
8. Member of operator (.)	9. Console Input/output
9. Constructor creation	10. Constructor call
10. Object oriented/language concepts	11. Object oriented/language syntax
11. Primitive data types/literals	12. Advanced data types
12. String literals	13. String variables
13. Concept of References	14. Implementation of pointers/references

Any Questions?

Operators in Java are similar to that of C++

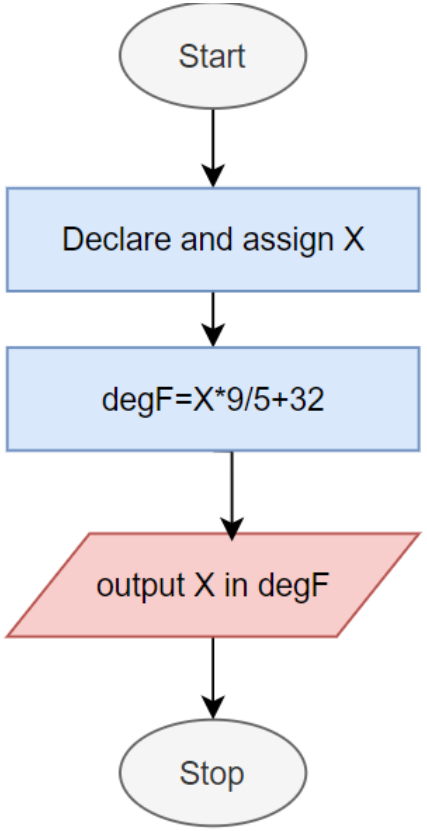
- Arithmetic Operators (+, -, /, *, %)
- Relational operators (==, !=, >,<,>=,<=)
- Assignment Operations (=, +=,-=, /=, *=, %=, ++, --)
- Logical operators (&&,||, !)
- Conditional (Ternary) Operator (?:).
- Bitwise operators (&, |, ^, ~, >>, <<)
- Use this resource for individual examples of the above

Any Questions?

Every program follows an algorithm

- Write a program given any real variable declared as X, can convert that X from the Celsius for Fahrenheit.

```
public class Main {
    public static void main(String[] args) {
        float x = 5;
        float degF = 9/5*x+32;
        System.out.println((x ) + " degC =  " + (degF) + " degF");
    }
}
```



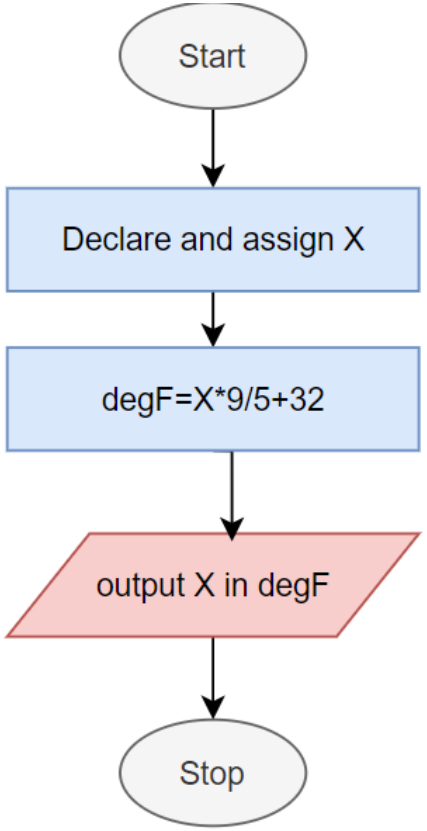
Static methods in Java

- Functions are Advanced operators (they take input and return an output mapping)
- In Java, Functions are called methods

- Methods are the basic Algorithmic Units of a program
- Method Concepts
  - Creating methods (composition)
    - Signature
      - Name
      - Return type
      - Type modifiers
      - Input parameters
    - Method body
      - Algorithmic contents
  - Using methods
    - Method declaration
    - Method definition
    - Method call
- What is a static method
  - Static methods can be called without an object instance variable
  - Private methods are called within the class and Public methods can be called outside the class

Static method example

```
public class Main {
    public static void main(String[] args) {
        float x = 5;
        float degF = degF(x);
        System.out.println((x ) + " degC =  " + (degF) + " degF");
    }
    public static float degF(float x) {
        float degF = 9/5*x+32;
        return degF;
    }
}
```



Any Questions?

Exercises

1. Write a simple Java program that prints “Hello from Java”
2. Write a program that gives the sum, product, subtraction and division of two numbers
3. Write a program that takes a real number variable X and converts the value from degrees Celsius to degrees Fahrenheit.

Advanced Questions

1. Write a program that takes a real number variable X and converts the value from degrees Fahrenheit to degrees Celsius.
2. Write seven static methods in a single Main class, and then, call them all from the main method displaying their results. The static methods should comprise all the above seven programs. One static method for each question. The static method for question 1 should be called “sayHello()” the arithmetic operator methods should be called “add()”, “sub()”, “mult()” and “div()” and the temperature methods should be called toCelsius() and toFahrenheit().