

PROGRAMMING PARADIGM-JAVA

Revision Lecture

COURSEWORK I

- Time Complexity Marking
- Use of Regex

```
while ((line = reader.readLine()) != null)
    String[] words = line.split("\\s+");
    String word = words[0];

    switch (word) {
        case "Iteration":
            case "Phase":
                if (words[i].equals("Ends"))
```

ASSESSMENT

- PGP Coursework 1 – Java (10%)
- PGP Coursework 2 – Haskell (15%)
- Exam (75%)
 - Time allowed: 2 hours and 30 minutes
 - Additional material: Haskell standard prelude

PGP EXAM

- Four questions:
 - Two Java questions (50%)
 - Two Haskell questions (50%)
- Type of questions:
 - Knowledge-based, e.g., conceptual questions...
 - Program-based, e.g., computational questions...
- Past exam papers: available on Moodle

PGP EXAM-JAVA

Which of these statements are true and which are false.

[4 marks]

- (i) An instance can be created from an abstract class.
- (ii) Multiple inheritance of interfaces is possible.
- (iii) A protected method in a class A can be accessed by a subclass of A.
- (iv) All classes and interfaces in Java are subclasses of the `Object` class.

Describe these two approaches to polymorphism in Java: Overloading and Overriding.
Give an example of each.

[8 marks]

Given the code, then:

```
(ii) System.out.println(l1.toString(10));
```

What will happen if the following code is run:-

```
System.out.println(l1.toString());
```

Write code using a `for` loop that takes an array `int[] ns` and will

JAVA CONTENT

- Properties of OOP:
 - Encapsulation:
 - Access modifiers
 - Package
 - Inheritance:
 - Superclass and subclass
 - Abstraction:
 - Abstract class and Interface
 - Polymorphism:
 - Overload and override

JAVA CONTENT

- Static vs non-static
- Override vs overload
- Abstract vs concrete
- Keywords: this, super
- Inner class: scope, name, creation, access modifier...
- Loops: for, while, foreach, label
- Recursive call in Java
- Generic data type

JAVA CONTENT

- Static vs non-static
- Override vs overload
- Abstract vs concrete
- Keywords: this, super
- Primitive type vs OO type: call by value, call by reference, data wrapper...
- Loops: for, while, foreach, label, break, continue ...
- If-then-else, switch
- Recursive call in Java
- Generic data type

JAVA CONTENT

- Inner class: scope, creation, access modifier...
- Input and output:
 - Stream: byte stream and character stream
 - BufferedReader and Scanner
 - FileReader and FileWriter
- Exceptions:
 - Checked vs unchecked
 - Try and catch
 - Throws