

## Assignment 1: KWIC

For this assignment, you will implement a KWIC (Key Word In Context) index system. You can refer to the readings available in the IVLE workbin/Readings folder.

1. "On the criteria to be used in decomposing systems into modules" by D L Parnas
2. "An Introduction to Software Architecture" by David Garlan and Mary Shaw

Choose **any two architectural designs** mentioned in the KWIC **case study-1** of Reading#2 mentioned above. Alternatively, you can modify any of the given design.

**Implement** your chosen (or modified) designs in Java.

Submit the **code** and a **short report** .

Follow the **description** and **submission instructions** given below.

## Description

KWIC (Key Word In Context) index system provides a search mechanism for information in a long list of lines, such as movie titles. Given a list of titles and a list of "words to ignore", you are to write a program that generates a KWIC (Key Word In Context) index of the titles. In a KWIC-index, a title is listed once for each keyword that occurs in the title. The keyword cannot be in "words to ignore" and the KWIC-index is alphabetized by keyword.

For each input title, it shall be "circularly shifted" exhaustively by removing the first word and appending it at the end of the line to create a set of circularly shifted lines. The first word (not including "words to ignore") will be the keyword. The system shall output on the screen a listing of the circularly shifted lines for all input titles in ascending alphabetical order.

For example, if the "words to ignore" are *is, the, of, and, as, a, after* and the titles are *The Day after Tomorrow, Fast and Furious, Man of Steel*, the output should be as below.

Day after Tomorrow the  
Fast and Furious  
Furious Fast and  
Man of Steel  
Steel Man of  
Tomorrow the Day after

The index system you implement should be:

- User friendly - the system user interface should be easy for users to use the system
- Response time - the system should respond to user's query in a reasonable time e.g. comparable to the common search engine such as Yahoo.
- Extendable - new functions can be added to the system easily without any major changes to the design of the system.

Should you want, you could include more criteria to the above for design and implementation.

**Submission Date/Time** **Tuesday 31 Jan 10am**

## **Submission Guidelines**

1. This is a paired task. Split the work between two members. Clearly state who implements which part in the report.
2. Submit your report(about 2-5 pages) as a single PDF file. Submit report in the folder A1-KWIC in IVLE Files(workbin). Exceeding the page guideline of 2-5 pages does not invite any penalty.
3. Name the document: "A1\_<Matric-number-1>\_<Matric-number-2>" - e.g. "A1\_A0045396X\_A0046342Y".
4. You will be required to demonstrate your work following the submission. A schedule will be posted near to the time. Each demonstration will take 5-10 minutes.
5. Use Git repository for the assignment. Push your codes to the repository and add your repository URL in the report you submit. You can sign up for a free account at <https://bitbucket.org> or <https://github.com>. If you are not sure how to use Git, take a look at [Pro Git book](#).
6. We provide a report template that you can use as in the next page.

## **Grading**

Assignment 1 is worth 5% of your final grade.

Grading will be based on the following:

- (i) How close the implementations are to the specified designs.
- (ii) How complete and informational report sections are.

*Late submission, incomplete work or missing demonstration could invite penalty.*

--- report template ---

## Assignment 1: KWIC

Code Repository URL:

<b>Name</b>		
<b>Matriculation Number</b>		

### 1. Introduction

up to 1 paragraph about the assignment. Specify who has done which implementation

### 2. Requirements

Specify functional and non-functional requirements against which you design and implement. Refer to lecture material of first two lectures to get information on requirement specification.

### 3. Architectural Design

Include a short description of the designs you have selected (or modified) to implement. Illustrate it using one or more diagrams e.g component diagram, class diagram. Use any diagramming notation you are familiar with. Label your diagrams appropriately.

### 4. Limitation & Benefits of Selected Designs

State limitation and benefits of the design you implemented.

### 5. Any other information

e.g. testing method you followed or test cases you designed or any other information