Assignment On

# Code Structure Measurement of SPL I

***Course Code**: SE 3104*
***Course Title:** Software Metrics Lab*
***Submission Date**:* 05.03.2023

## Submitted To:

Mr. Dipok Chandra Das
Assistant Professor, IIT
Noakhali Science and Technology University (NSTU)

## Submitted By:

| | |
|---|---|
| Md. Al-Amin | ASH1925008M |
| Abdullah Alif | ASH1925009M |
| Ayesha Nasrin Ripa | BKH1925025F |

**Noakhali Science and Technology University**

## Table of Contents

1. Code Structure Measurement: ..... 3

2. Types of Measurement: ..... 3

3. Cyclomatic Complexity Measurement: ..... 3

4. Cyclomatic Complexity of Each Class: ..... 4

Book_pdf_frame.java ..... 4

Main_frame.java ..... 4

Main.java ..... 5

Download.java ..... 5

Download_Without_saving.java ..... 6

View.java ..... 7

Teacher_info.java ..... 8

Search_item.java ..... 9

Slides.java ..... 10

File_Basic.java ..... 11

Offline_pdf.java ..... 12

Home_frame.java ..... 13

Welcome_frame.java ..... 14

## List Of Figures:

Figure 1: Book_pdf_frame.java DD graph ..... 4

Figure 2:main_frame.java DD graph ..... 4

Figure 3:main.java DD graph ..... 5

Figure 4:Download.java DD graph ..... 5

Figure 5:Download_Without_saving.java DD graph ..... 6

Figure 6:view.java DD graph ..... 7

Figure 7:Teacher_info.java DD graph ..... 8

Figure 8: search_item.java DD graph ..... 9

Figure 9:Slides.java DD graph ..... 10

Figure 10:File_Basic.java DD graph ..... 11

Figure 11:Offline_pdf.java DD graph ..... 12

Figure 12:Home_frame.java DD graph ..... 13

Figure 13:Welcome_frame.java DD graph ..... 14

# 1. Code Structure Measurement:

The estimation of a software's structural qualities is crucial for both the product's maintenance and the development effort. Understanding the complexity involved in changing one product into another, testing a product, or forecasting the external software attributes from early internal product measures is made easier by looking at the structure of requirements, design, and code.

# 2. Types of Measurement:

The structure of software can be measure using

- **Control-flow structure** − It is the sequence in which instructions are executed in a program.
- **Data-flow structure** − It is the behavior of the data as it interacts with the program.

We have used Cyclomatic Complexity measurement to understand the complexity of our system.

# 3. Cyclomatic Complexity Measurement:

The Cyclomatic complexity defines the number of independent paths in the basis set of the program that provides the upper bound for the number of tests that must be conducted to ensure that all the statements have been executed at least once.

There are three methods of computing Cyclomatic complexities.

**Method 1**: Total number of regions in the flow graph is a Cyclomatic complexity.

**Method 2**: The Cyclomatic complexity, V (G) for a flow graph G can be defined as

$$V(G) = E - N + 2$$

Where: E is total number of edges in the flow graph. N is the total number of nodes in the flow graph.

**Method 3**: The Cyclomatic complexity V (G) for a flow graph G can be defined as

$$V(G) = P + 1$$

Where: P is the total number of predicate nodes contained in the flow G.

# 4. Cyclomatic Complexity of Each Class:

## Book_pdf_frame.java

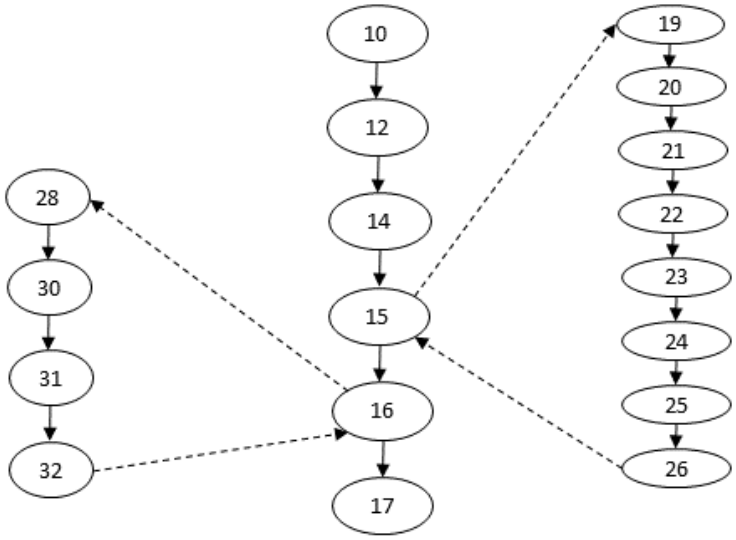


*Figure 1: Book_pdf_frame.java DD graph*

**Complexity**: 3

## Main_frame.java

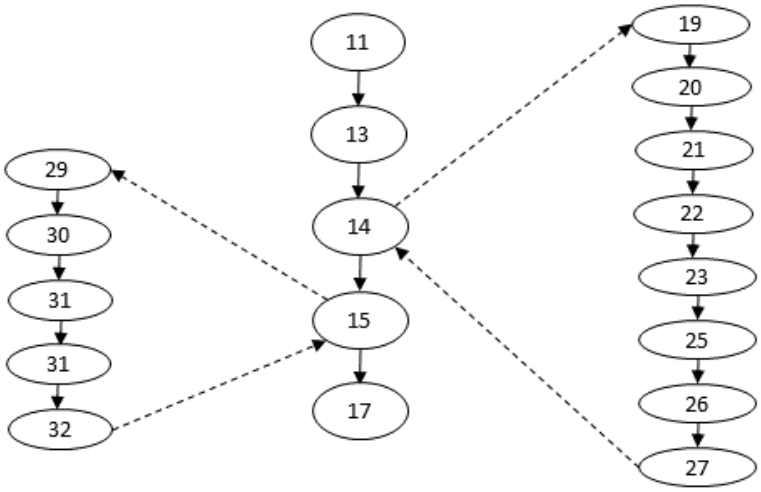


*Figure 2:main_frame.java DD graph*

**Complexity**: 3

**Main.java**

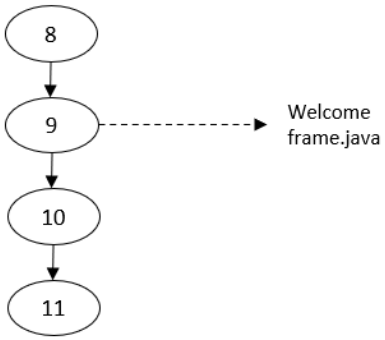


***Figure 3:main.java DD graph***

**Complexity**: 1

**Download.java**

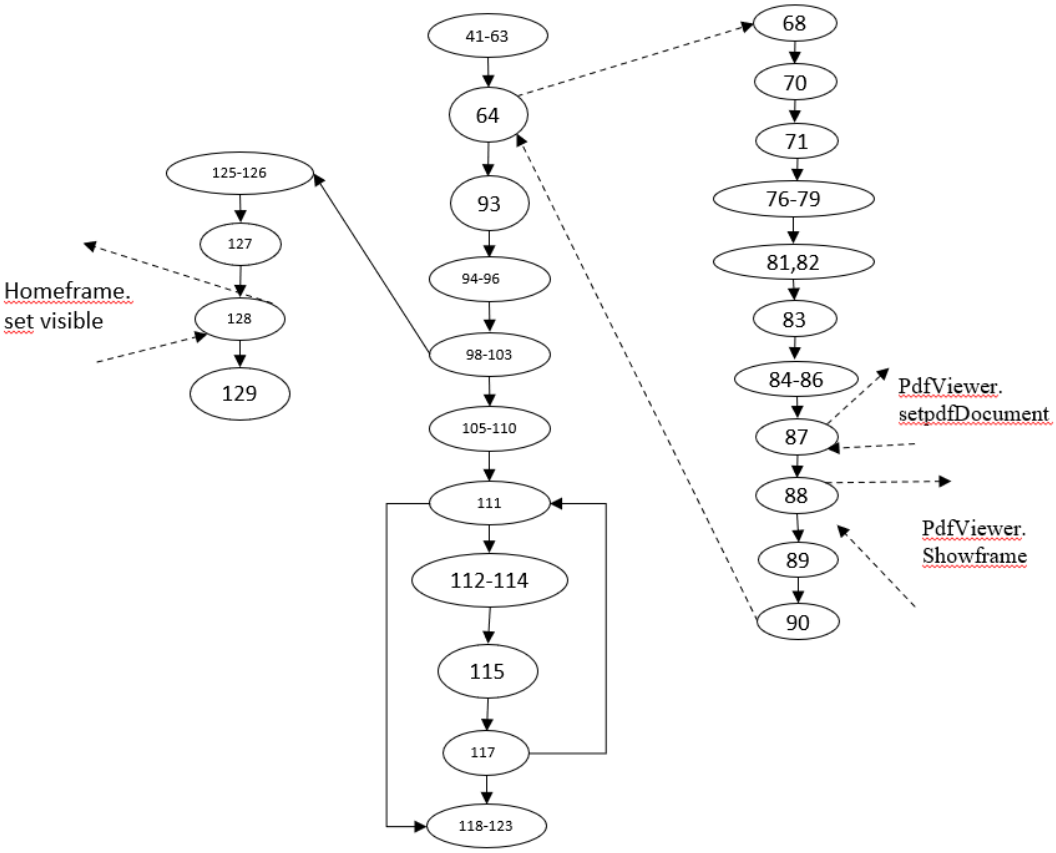


***Figure 4:Download.java DD graph***

**Complexity**: 8

**Download_Without_saving.java**

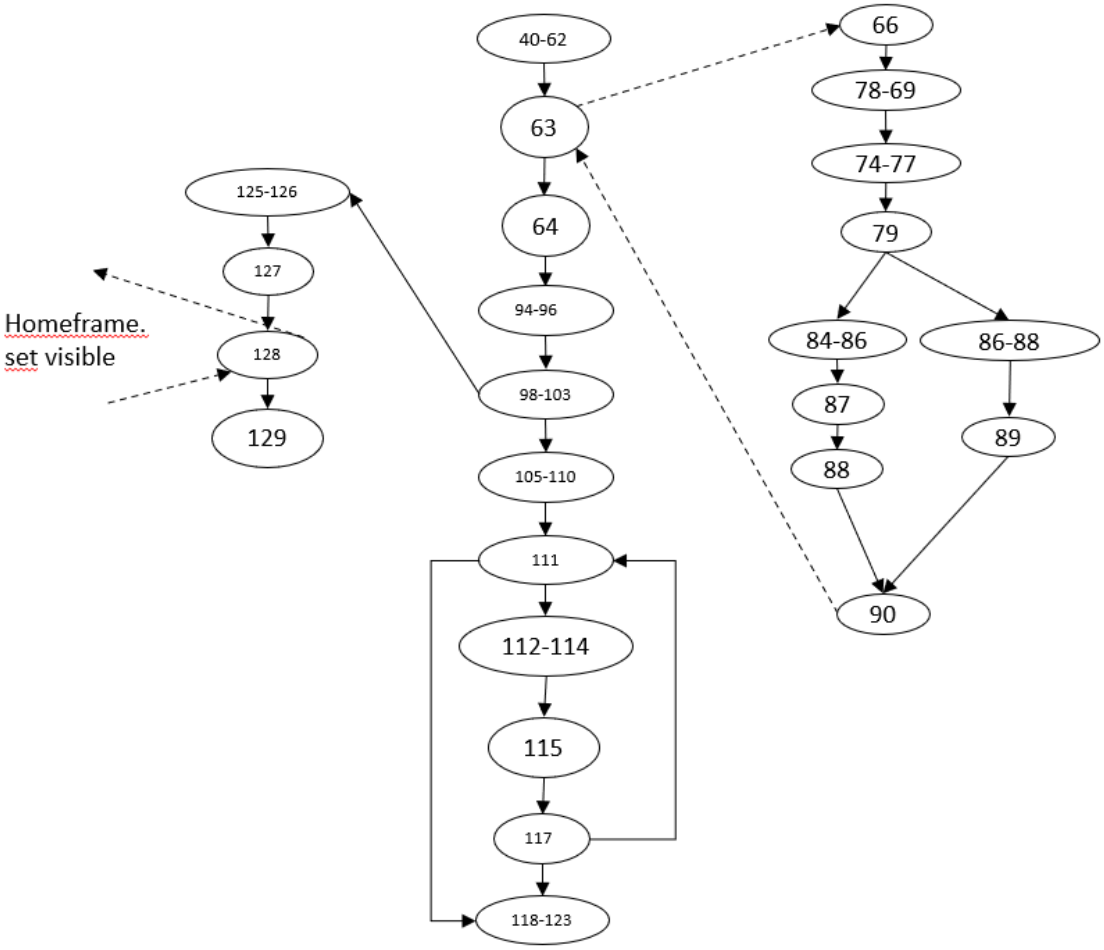


*Figure 5:Download_Without_saving.java DD graph*

**Complexity**: 11

**View.java**

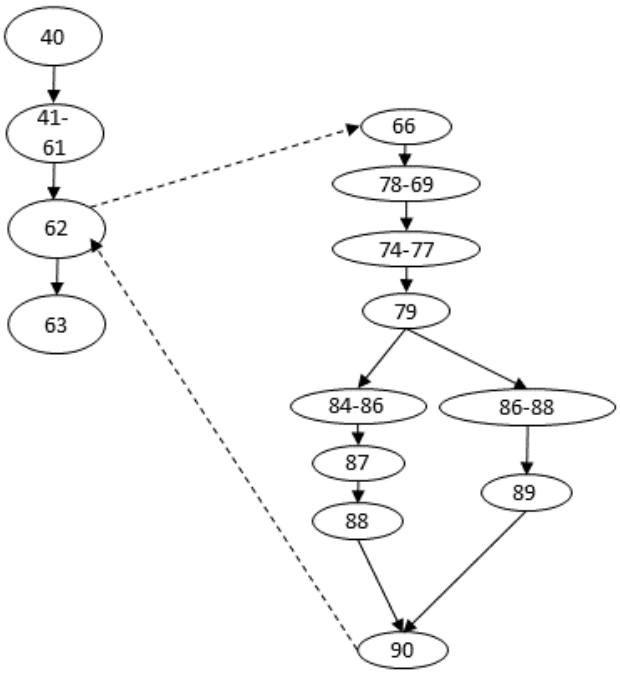


***Figure 6:view.java DD graph***

**Complexity**: 8

**Teacher_info.java**

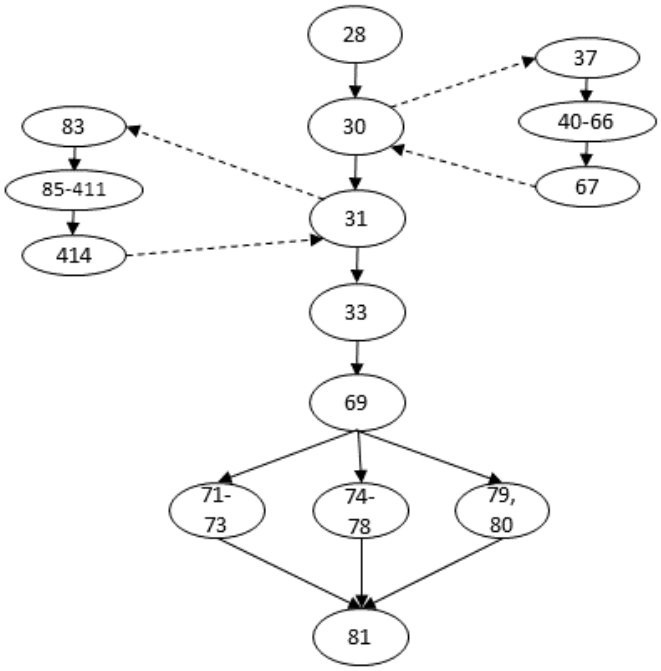


***Figure 7:Teacher_info.java DD graph***

**Complexity**: 6

**Search_item.java**

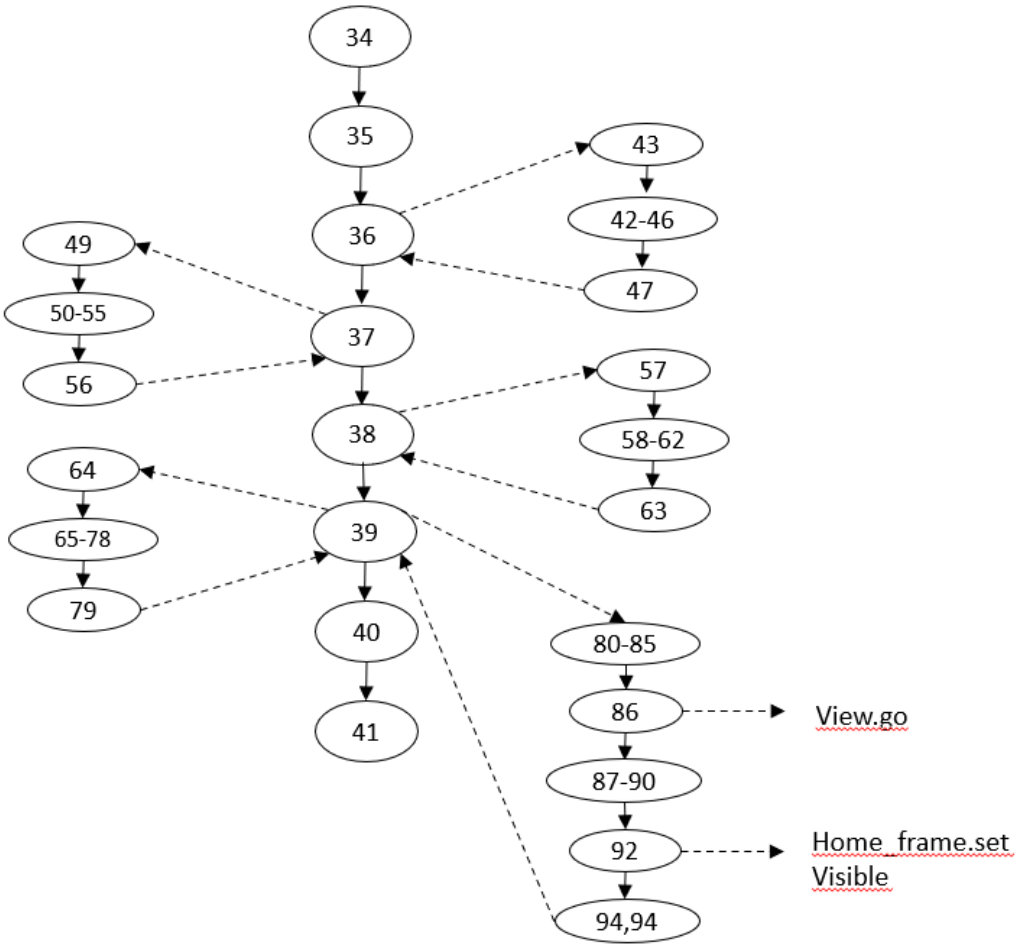


*Figure 8: search_item.java DD graph*

**Complexity**: 13

**Slides.java**

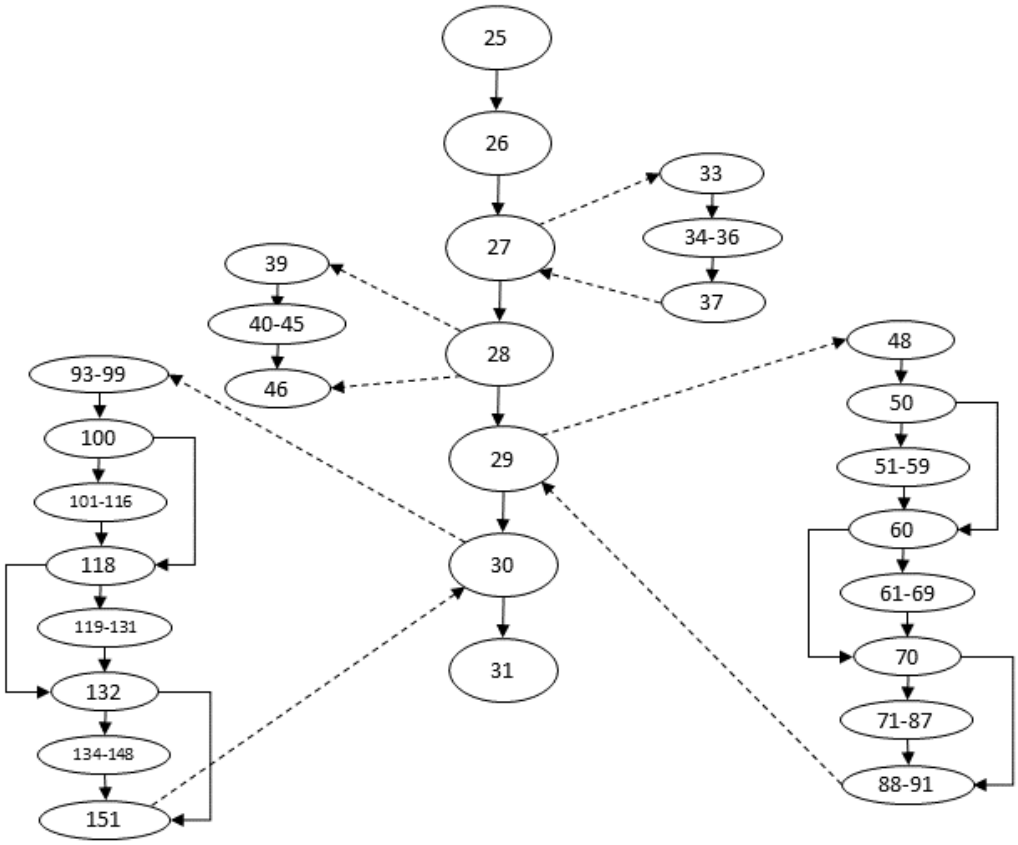


*Figure 9:Slides.java DD graph*

**Complexity**: 18

**File_Basic.java**

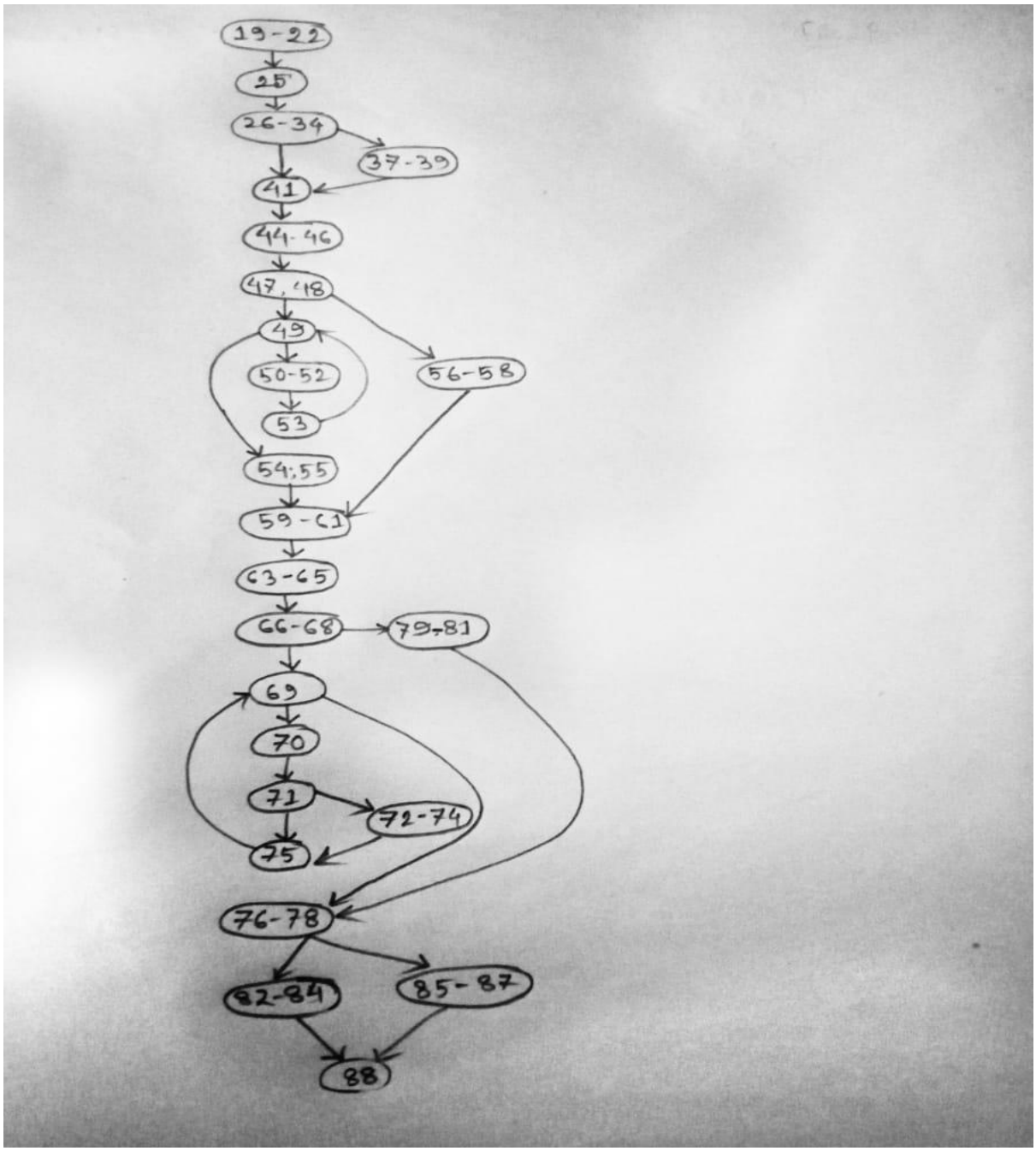


*Figure 10:File_Basic.java DD graph*

**Complexity:** 7

**Offline_pdf.java**

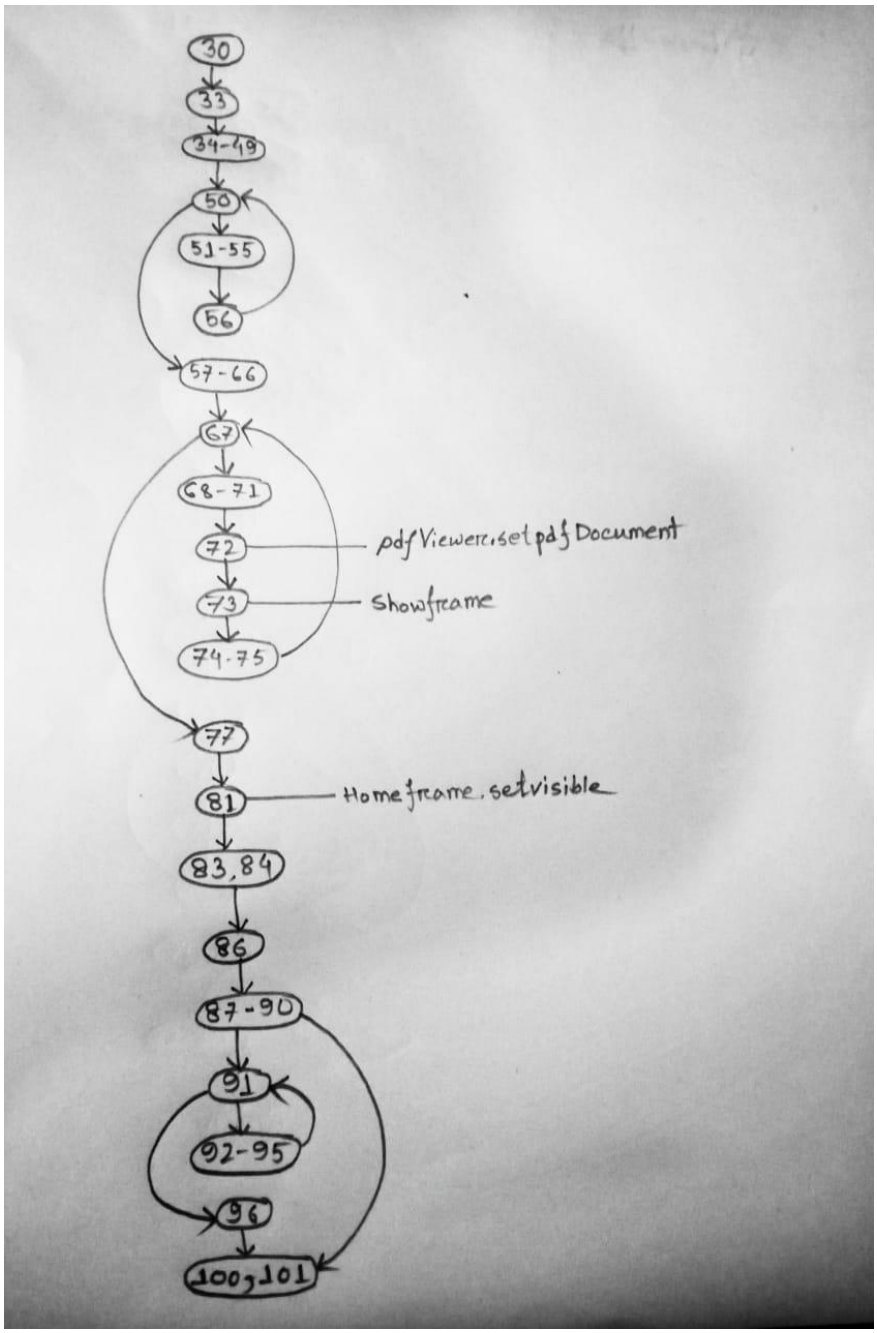


*Figure 11:Offline_pdf.java DD graph*

**Complexity**: 8

**Home_frame.java**

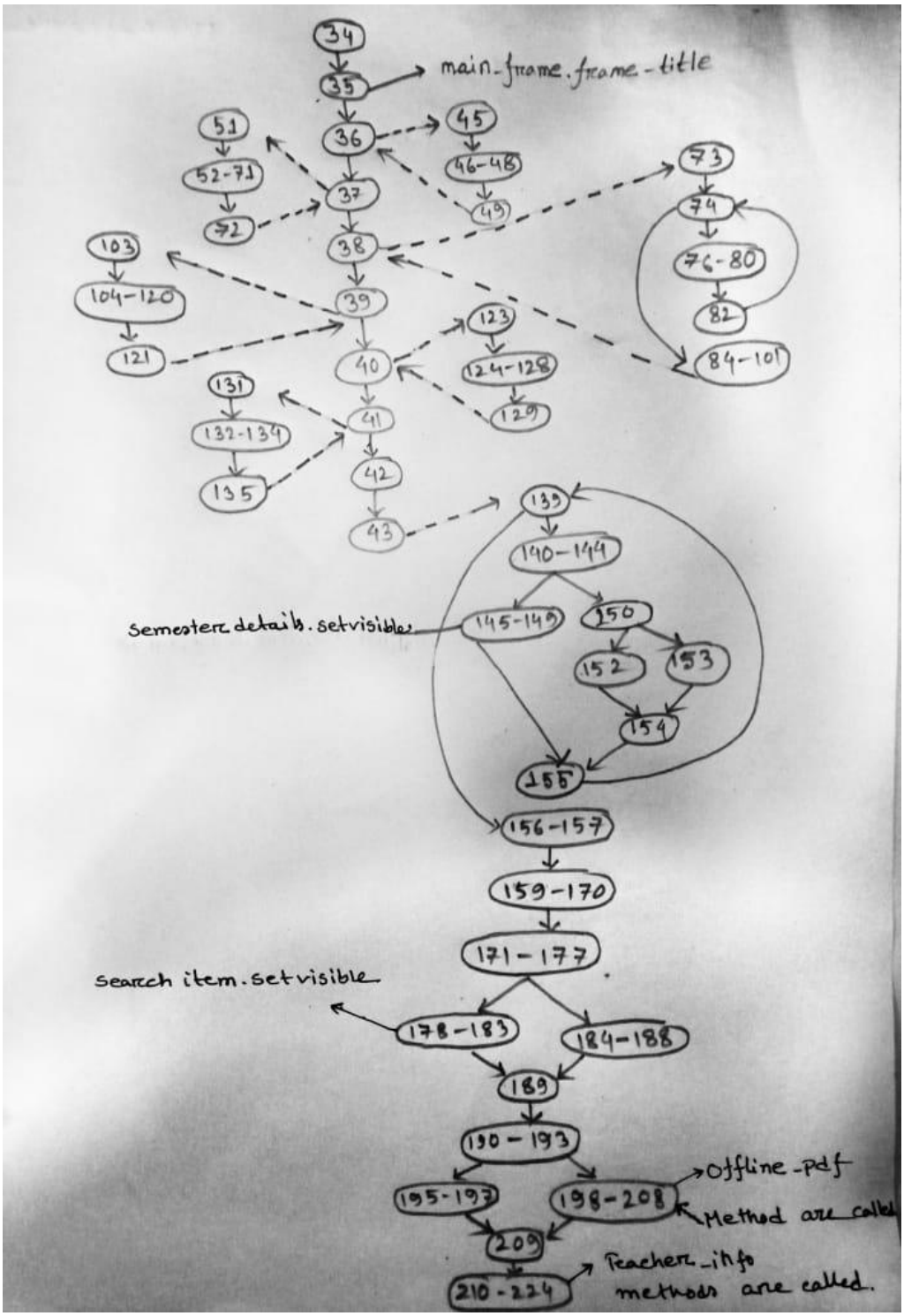


*Figure 12:Home_frame.java DD graph*

**Complexity**: 21

**Welcome_frame.java**

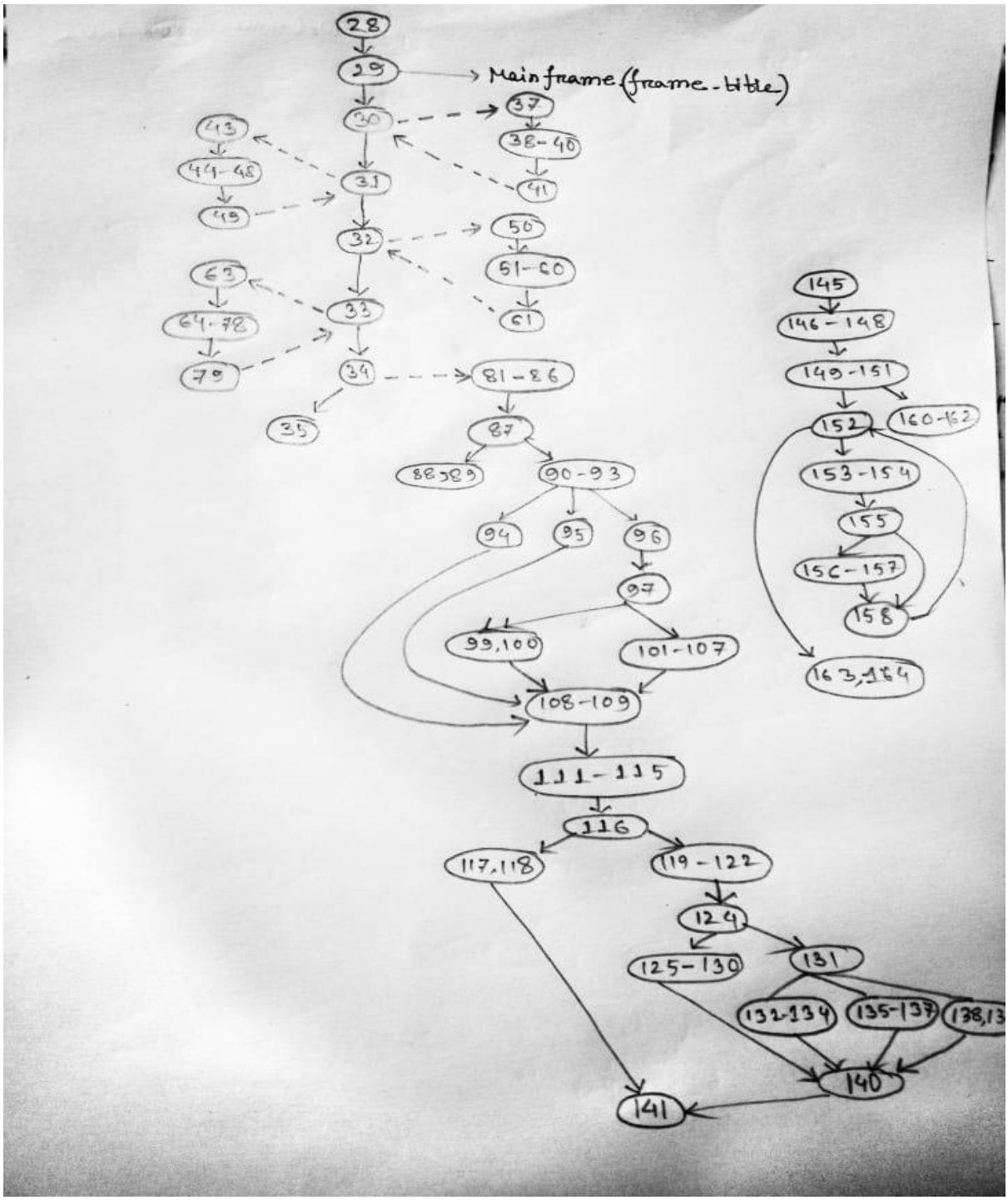


*Figure 13:Welcome_frame.java DD graph*

**Complexity**: 23