

Assignment On Object-Oriented 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

Abdullah Alif

Ayesha Nasrin Ripa

ASH1925008M

ASH1925009M

BKH1925025F



Noakhali Science and Technology University

Table of Contents

Object-Oriented Structure Measurement:	3
1. Measuring Coupling:	3
1.1 Coupling Between Objects (CBO).....	3
1.2 Response for Classes (RFC)	5
1.3 Message Passing Coupling (MPC)	6
2. Measuring Cohesion:	7
2.1 Lack of Cohesion Metric (LCOM)	7
3. Object Oriented Length Measures:	8
3.1 Depth of Inheritance Tree (DIT).....	8
4. Object Oriented Reuse Measures:.....	9
4.1 Number of Children (NOC)	9

List of Table:

Table 1: CBO of Each Class	3
Table 2: RFC of Each Class	5
Table 3: MPC of Each Class	6
Table 4: LCOM of Each Class.....	7
Table 5: DIT of Each Class.....	8
Table 6: NOC of Parent Classes	9

List Of Figure:

Figure 1: Class Coupling Metric	4
Figure 2: Inheritance Tree.....	8

Object-Oriented Structure Measurement:

Structural properties that capture inter-connections among classes are believed to be important to measure. This is because they are considered to affect cognitive complexity. Object-oriented metrics measure these structural properties. Coupling metrics characterize the static usage dependencies among the classes in an object-oriented system. Cohesion metrics characterize the extent to which the methods and attributes of a class belong together. In addition, inheritance is also believed to play an important role in the understandability of object-oriented applications.

1. Measuring Coupling:

1.1 Coupling Between Objects (CBO)

Description: Represents the number of classes coupled to a given class. Coupling can occur through method calls, field accesses, inheritance, arguments, return types, and exceptions.

Measurement Procedure: Count the number of those followings: -

- Method call
- Class extends
- Properties or parameters
- Method arguments or return type
- Variables in methods

Values:

Class Name	CBO
Main_frame	1
book_pdf_frame	4
main	0
Download	1
Download_without_saving	0
File_basic	2
Home_frame	10
offlinepdf	1
pdf_book	4
pdfViewer	4
search_item	1
Semester_details	3
Slide_details	0
Slides	1
Teacher_info	1
view	3
welcome	1

Table 1: CBO of Each Class

	main_frame	book_pdf_frame	main	Download	Download_without_sav	File_basic	Home_frame	offlinepdf	pdf_book	pdfViewer	search_item	Semester_details	Slide_details	Slides	Teacher_info	view	welcome
main_frame																	
book_pdf_frame																	
main																	Object
Download						Object	Object		Variable	Object							
Download_without_saving							object			object							
File_basic									Variable								
Home_frame								Object			object	Object			object		
offlinepdf							Object			Object							
pdf_book				Variable		Object	Object			Object		Variable					Instance
pdfViewer							Object										
search_item		Extends					Object										Instance
Semester_details		Extends					Object		Object					Object			
Slide_details		Extends					Object										Instance
Slides		Extends															
Teacher_info							Object										
view										Object							
welcome	Extends						Object					Variable					
CBO	1	4	0	1	0	2	10	1	4	4	1	3	0	1	1	3	1

Figure 1: Class Coupling Metric

1.2 Response for Classes (RFC)

Description: The total number of methods that can potentially be executed in response to a message received by an object of a class.

Measurement Procedure:

- Identify the entry points or events that can trigger the execution of code in the system.
- For each entry point, identify all the classes that are directly or indirectly called during its execution.
- Count the total number of unique classes that are called by each entry point. This is the RFC for that entry point.
- Repeat steps 2 and 3 for all the entry points in the system.
- Sum the RFC values for all the entry points to obtain the total RFC value for the system.

Values:

Class Name	Number of Methods	Number of Methods called	RFC
main_frame	2	0	2
book_pdf_frame	2	0	2
main	1	1	2
Download	5	4	9
Download_without_saving	5	3	8
File_basic	3	0	3
Home_frame	8	13	21
offlinepdf	5	3	8
pdf_book	8	5	13
pdfViewer	14	1	15
search_item	9	2	11
Semester_details	8	3	11
Slide_details	6	3	9
Slides	7	1	8
Teacher_info	4	1	5
view	5	2	7
welcome	6	1	7

Table 2: RFC of Each Class

1.3 Message Passing Coupling (MPC)

Description: Measures the number of method calls defined in methods of a class to methods in other classes, and therefore the dependency of local methods to methods implemented by other classes.

Measurement Procedure:

- Identify the modules or components in the system that communicate with each other using message passing.
- For each module, count the number of different messages that it sends to other modules.
- For each module, count the number of different messages that it receives from other modules.
- Calculate the MPC value for each module as the sum of its in-degree and out-degree.
- Calculate the total MPC value for the system as the sum of the MPC values for all modules.

Values:

Class Name	MPC
main_frame	0
book_pdf_frame	0
main	1
Download	4
Download_without_saving	3
File_basic	0
Home_frame	13
offlinepdf	3
pdf_book	5
pdfViewer	1
search_item	2
Semester_details	3
Slide_details	3
Slides	1
Teacher_info	1
view	2
welcome	1

Table 3: MPC of Each Class

2. Measuring Cohesion:

2.1 Lack of Cohesion Metric (LCOM)

Description: LCOM is a measure for the number of not connected method pairs in a class representing independent parts having no cohesion.

Measurement Procedure: Automated Program

Values:

Class Name	LCOM
main_frame	0
book_pdf_frame	0
main	-1
Download	0
Download_without_saving	-1
File_basic	0
Home_frame	0.4
Offlinepdf	0.4
Pdf_book	0.117647058823529
PdfViewer	0.130434782608695
search_item	0.2
Semester_details	0.333333333333333
Slide_details	0.0625
Slides	0.666666666666666
Teacher_info	0
view	-1
welcome	0

Table 4: LCOM of Each Class

3. Object Oriented Length Measures:

3.1 Depth of Inheritance Tree (DIT)

Description: Depth of Inheritance Tree (DIT) is the maximum length of a path from a class to a root class in the inheritance structure of a system. DIT measures how many super-classes can affect a class.

Measurement Procedure: Calculate the depth of a class from root class.

Values:

Class Name	DIT
main_frame	0
book_pdf_frame	0
search_item	1
Semester_details	1
Slide_details	1
Slides	1
welcome	1

Table 5: DIT of Each Class

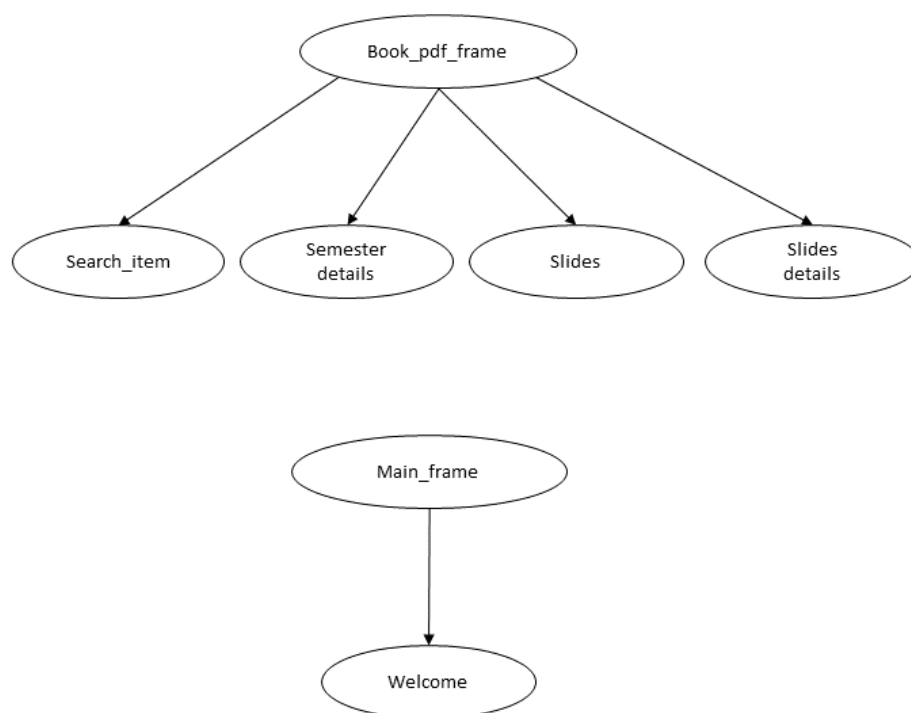


Figure 2: Inheritance Tree

4. Object Oriented Reuse Measures:

4.1 Number of Children (NOC)

Description: The number of children is the number of immediate subclasses subordinate to a class in the hierarchy. It is an indicator of the potential influence a class can have on the design and on the system

Measurement Procedure: Count the number of children a parent class has.

Values:

Class Name	NOC
main_frame	1
book_pdf_frame	4

Table 6: NOC of Parent Classes