

# (Object-Oriented Structure Measurement)

Course Code : SE 3204  
Course Title : Software Metrics

## **Submitted To:**

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

## **Submitted By:**

Ikra Chowdhury Nowkshi  
ID: BFH1925019F  
Bachelor of Science in Software Engineering

## **Team Members:**

- |                           |             |
|---------------------------|-------------|
| 1. Ikra chowdhury Nowkshi | BFH1925019F |
| 2. Ishrat Jahan Rintu     | BFH1925002F |
| 3. Roichuddin Rana        | ASH1925003M |
| 4. Dhruva Kanti Bakshi    | ASH1825018M |



# **Noakhali Science and Technology**

## Table of Contents

<b>1. Measuring Coupling in Object-Oriented System</b>	<b>3</b>
<b>1.1 Coupling between Object Class</b>	<b>3</b>
<b>1.2 Response for Class</b>	<b>5</b>
accountdata: 10	5
chatHistory: 4	6
objectread: 5	7
createaccountWindow: 8	8
FileShow: 3	9
loginwindow: 6	9
ObjectFileCreate: 2	10
personalchatroomui: 13	11
DeleteListedWord: 1	12
Dictionary: 8	13
DictionaryMainFrame: 13	14
<b>1.3 Message Passing Coupling</b>	<b>15</b>
accountdata: 4	15
chatHistory: 0	15
objectread: 4	16
createaccountWindow: 3	16
FileShow: 0	17
loginwindow: 2	17
personalchatroomui: 1	17
ObjectFileCreate:	17
DeleteListedWord:	17
Dictionary:	17
DictionaryMainFrame:	17
<b>2. Measuring Cohesion in Object-Oriented System</b>	<b>18</b>
<b>3. Object-Oriented Length Measure</b>	<b>21</b>
<b>4. Object-Oriented Reuse Measurement</b>	<b>22</b>

# 1. Measuring Coupling in Object-Oriented System

Coupling is a term used in software engineering to describe the degree to which different components or modules of a software system are interconnected or dependent on each other. In other words, coupling measures how much one part of the system depends on another part of the system to function correctly.

We can measure coupling in object-oriented system by

1. Coupling Between Object Class (CBO)
2. Response For Class (RFC)
3. Message Passing Coupling (MPC)

## 1.1 Coupling between Object Class

In object-oriented programming, coupling refers to the degree of interdependence between classes or objects. There are several types of coupling that can occur between object classes.

**Measurement Process:** We consider here

1. Methods Call
2. Class Extends

*Table 1: Coupling between Object*

Class	Number	Coupled With
userstructure	0	
accountdata	1	userstructure
chathistory	0	userstructure
objectread	1	userstructure
FileShow	0	
ObjectFileCreate	0	
createaccountWindow	4	ObjectFileCreate ButtonSound accountdata userstructure loginwindow
loginwindow	4	objectread personalchatroomui aaccountdata createaccountWindow

personalchatroomui	6	Chathistory Clintthread FileShow TictacMainFrame DictionaryMainFrame
serverroom	0	
TicTacToy	0	
TictacMainFrame	1	TicTacToy
clintsocket	0	
clintthread	0	
file_name_transfer	0	
clinthandler2	1	serversocket
serversocket	1	clinthandler2
AgeCal	0	
Converter	0	
DeleteListedWord	0	
Dictionary	0	
DictionaryMainFrame	5	ImplementWord DeleteListedWord WordSearch DisplayWord Dictionary
DisplayWord	0	
ImplementWord	0	
WordSearch	0	

## 1.2 Response for Class

In object-oriented programming, the response for a class refers to the actions or behavior that a class can perform in response to inputs or events. The level of coupling between a class and other classes in the system can affect its response, with low coupling promoting modularity, reusability, and maintainability. Good software design aims for low coupling and clear separation of concerns between classes.

**Measurement Process:** we count here number of methods in the class and number of distinct method calls made by the methods of this call.

**RFC** = number of methods in the class + number of distinct method calls made by the methods.

accountdata: 10

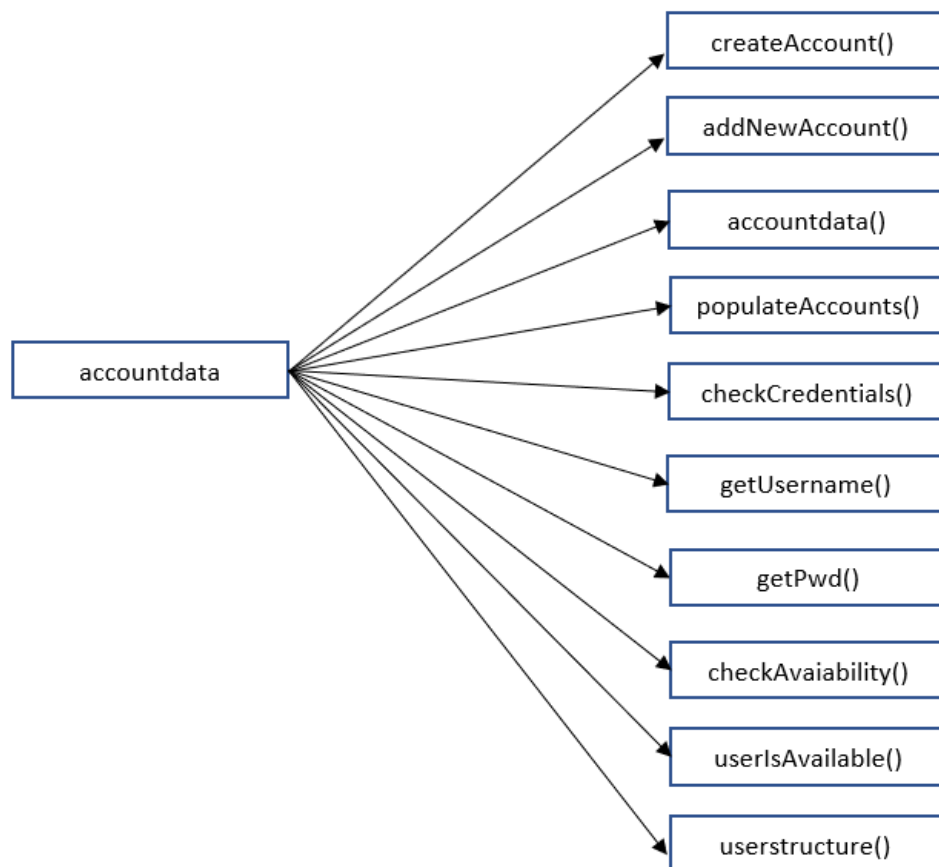
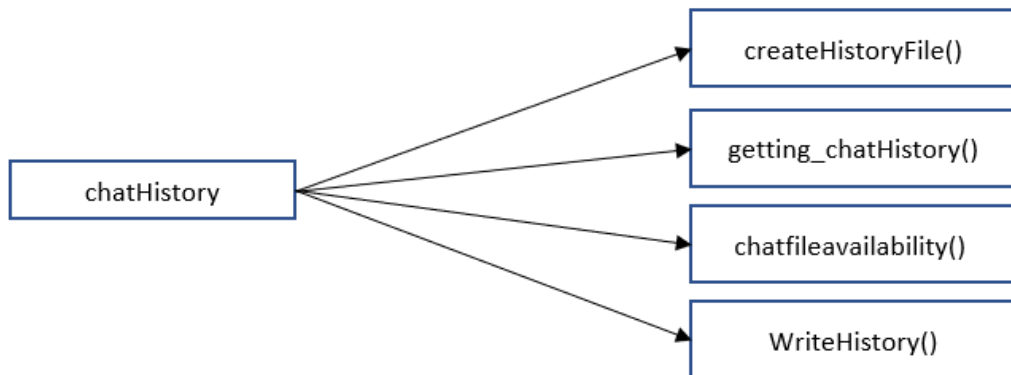


Figure 1: RFC (accountdata)

chatHistory: 4



*Figure 2: RFC (chatHistory)*

objectread: 5

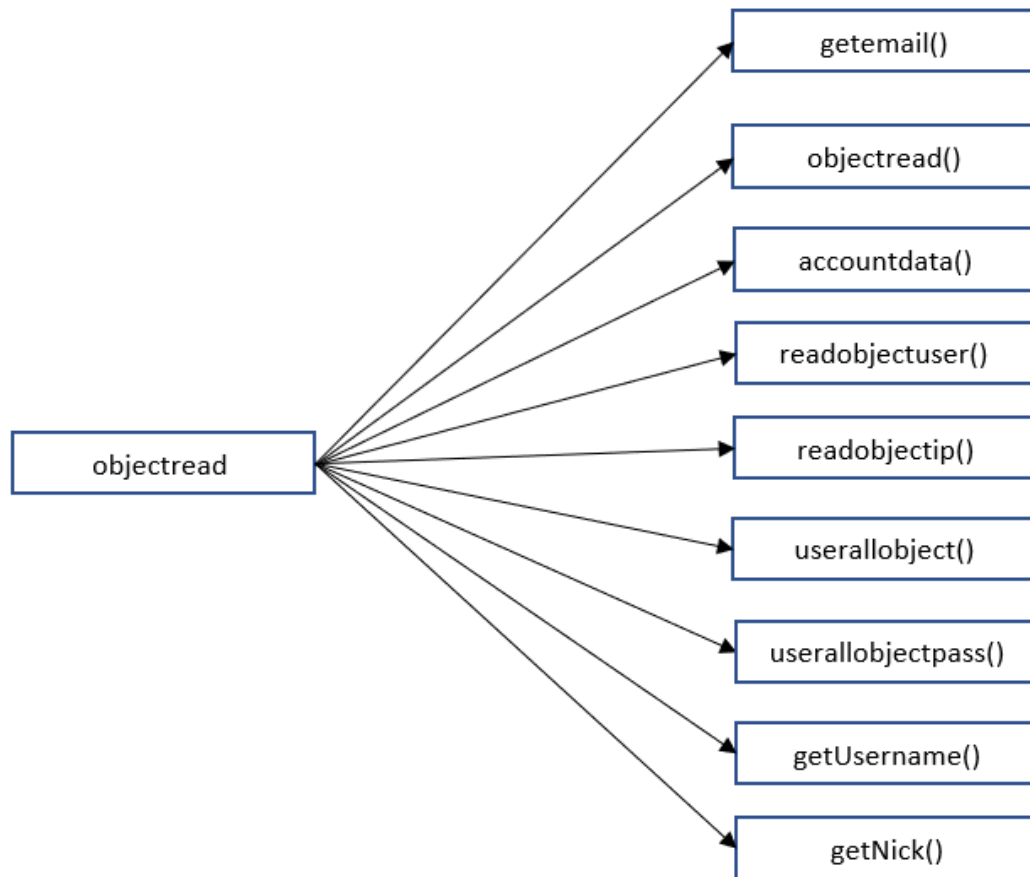


Figure 3: RFC (*objectread*)

createaccountWindow: 8

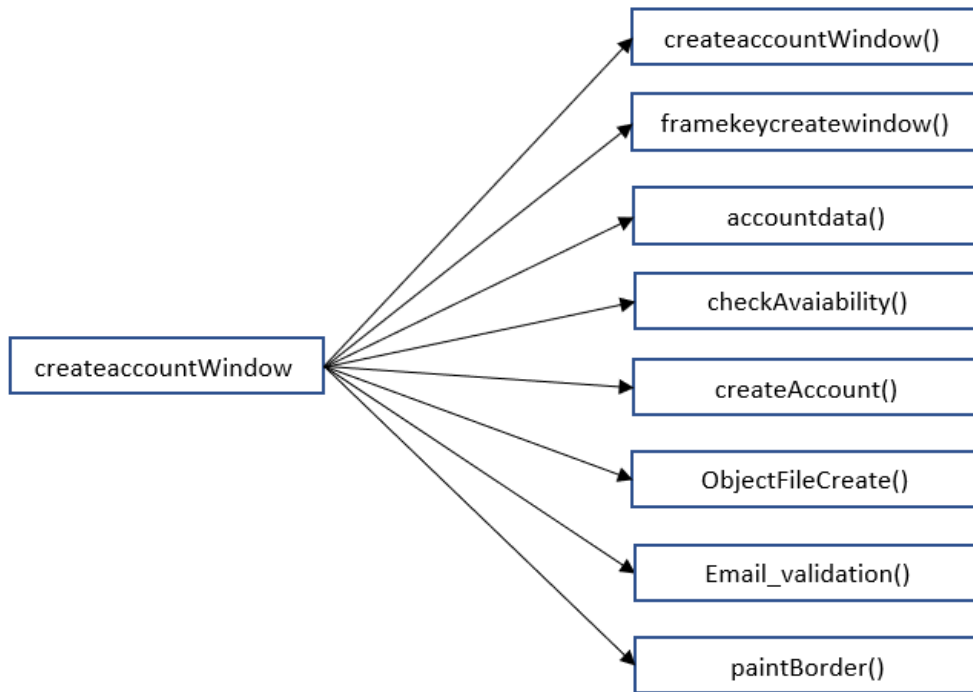


Figure 4: RFC (createaccountWindow)



FileShow: 3

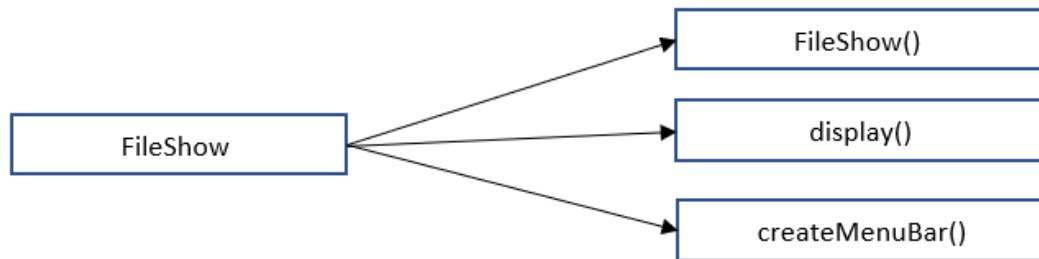


Figure 5: RFC (FileShow)

loginwindow: 6

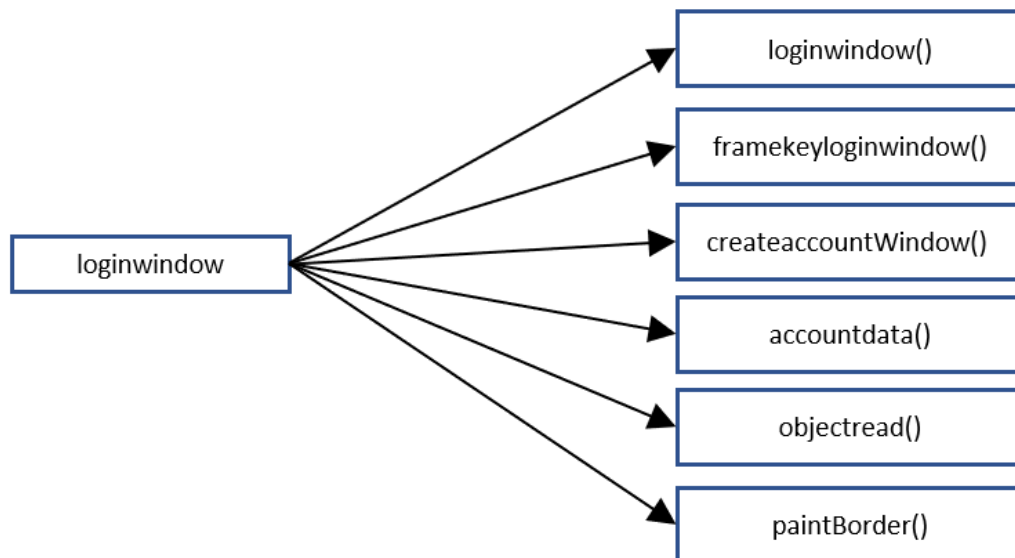
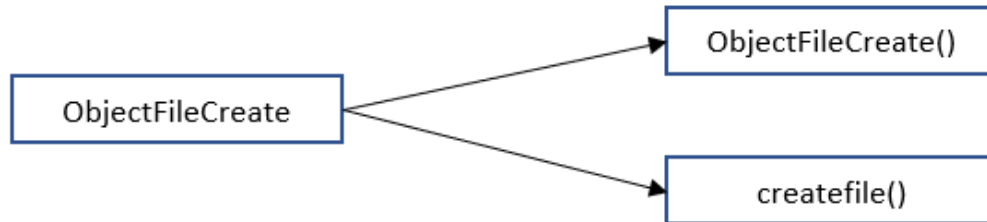


Figure 6: RFC (loginwindow)

ObjectFileCreate: 2



*Figure 7: RFC (ObjectFileCreate)*

personalchatroomui: 13

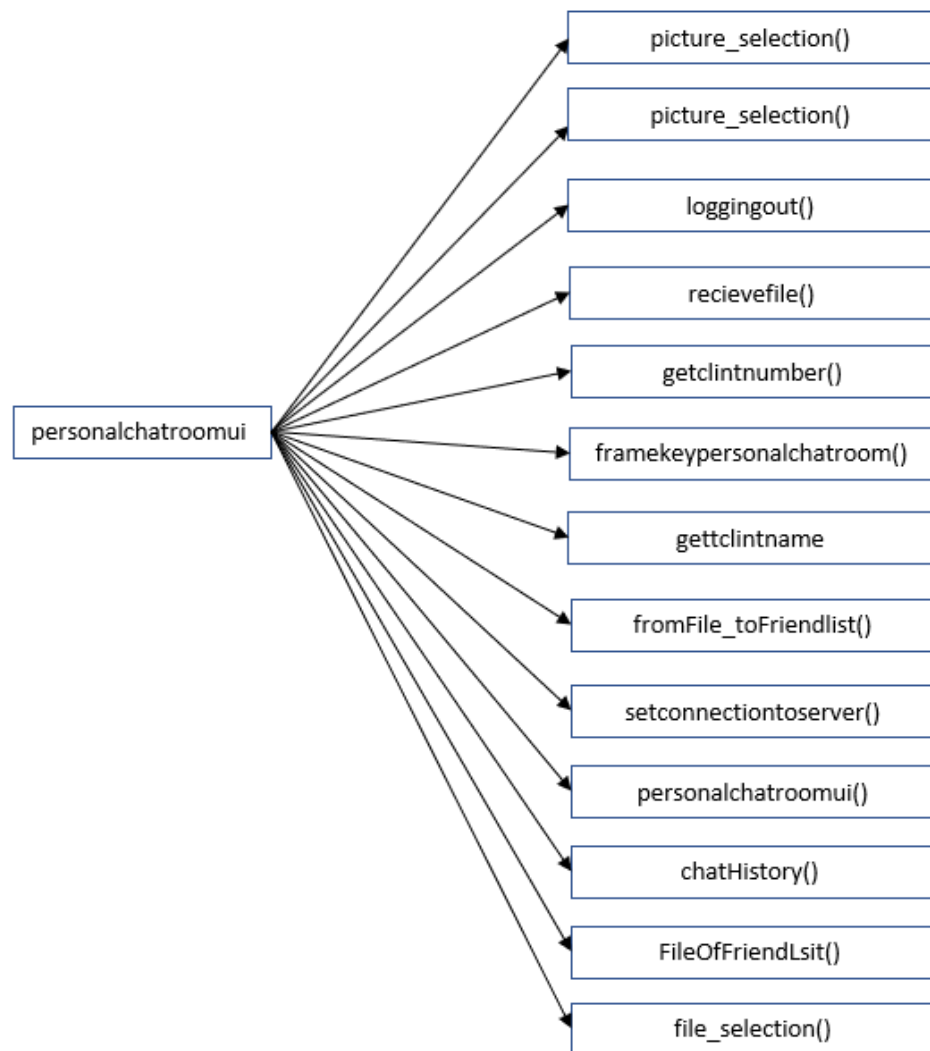
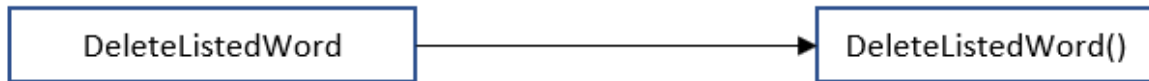


Figure 8: RFC (*personalchatroomui*)

DeleteListedWord: 1



*Figure 9: RFC (DeleteListedWord)*

Dictionary: 8

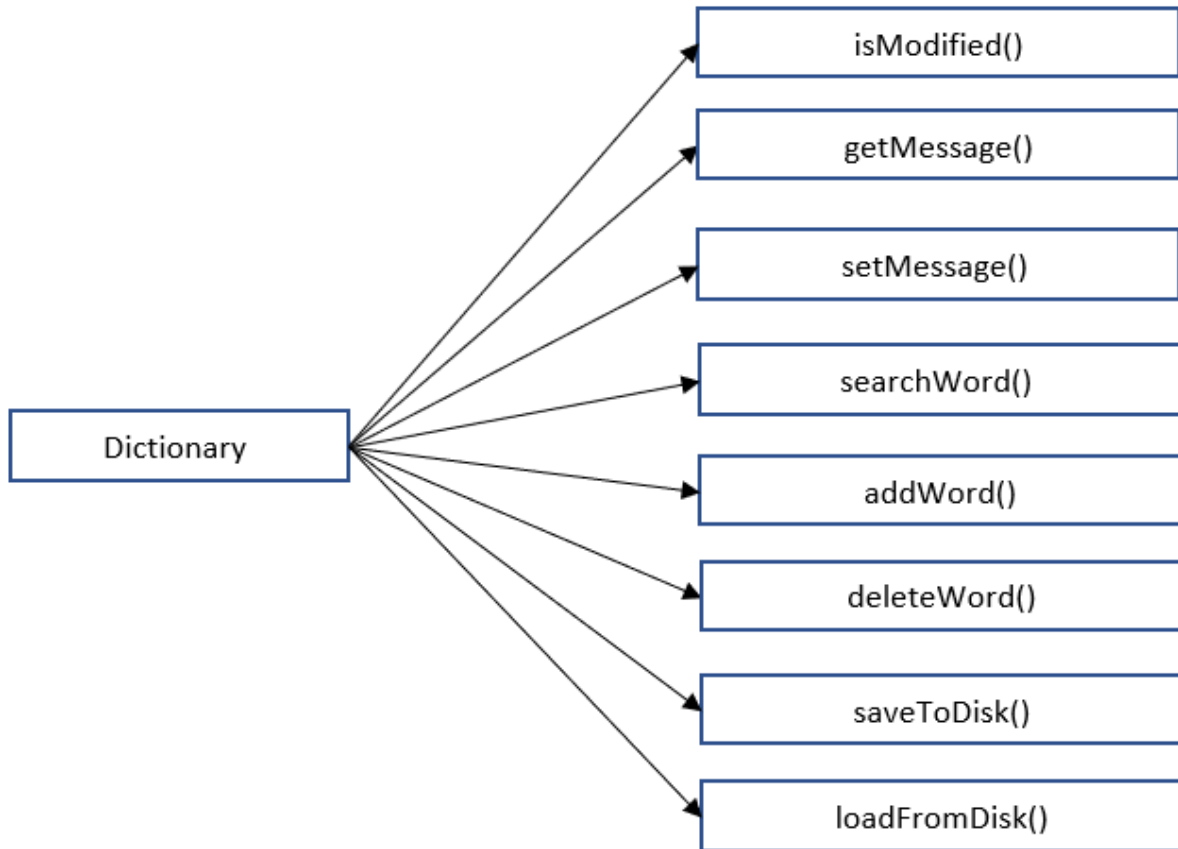


Figure10: RFC (Dictionary)

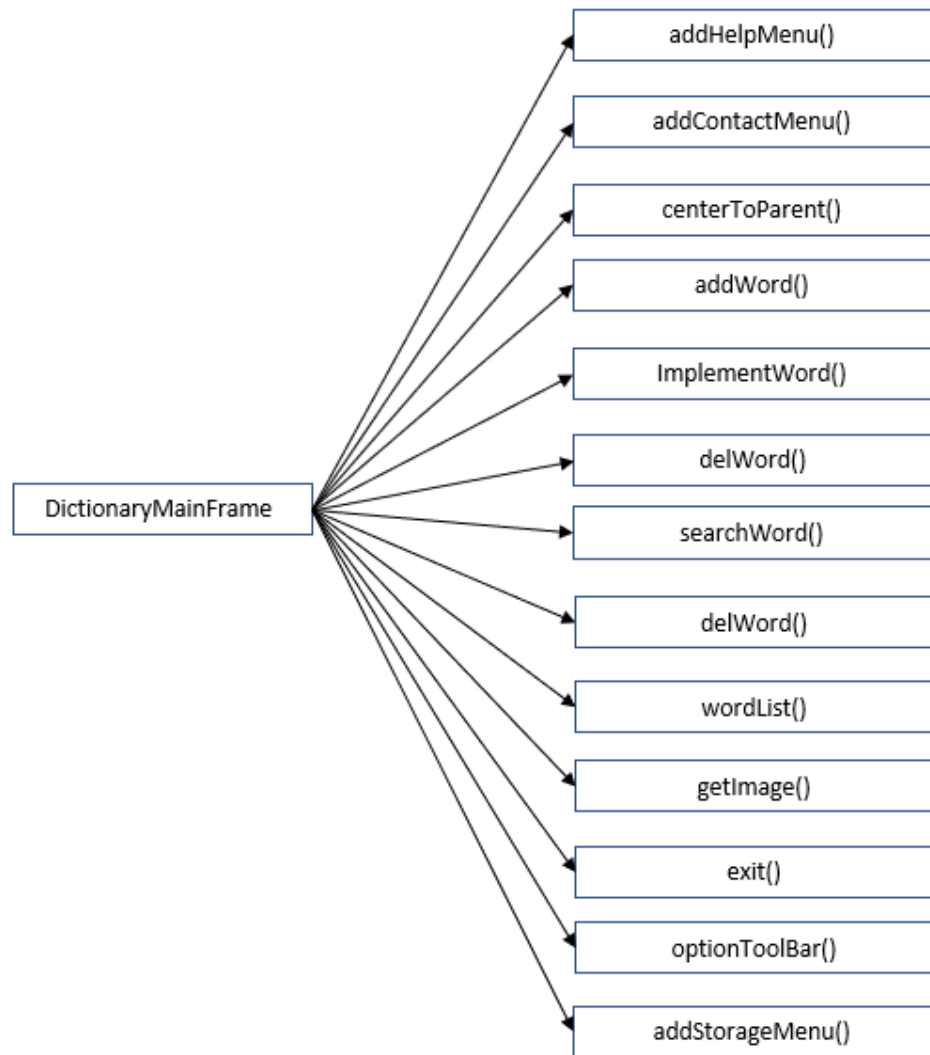


Figure 10:RFC (DictionaryMainFrame)

### 1.3 Message Passing Coupling

In object-oriented programming, the response for a class refers to the actions or behavior that a class can perform in response to inputs or events. The level of coupling between a class and other classes in the system can affect its response, with low coupling promoting modularity, reusability, and maintainability. Good software design aims for low coupling and clear separation of concerns between classes

Measurement Process: Number of distinct method calls made by the methods of this call.

accountdata: 4

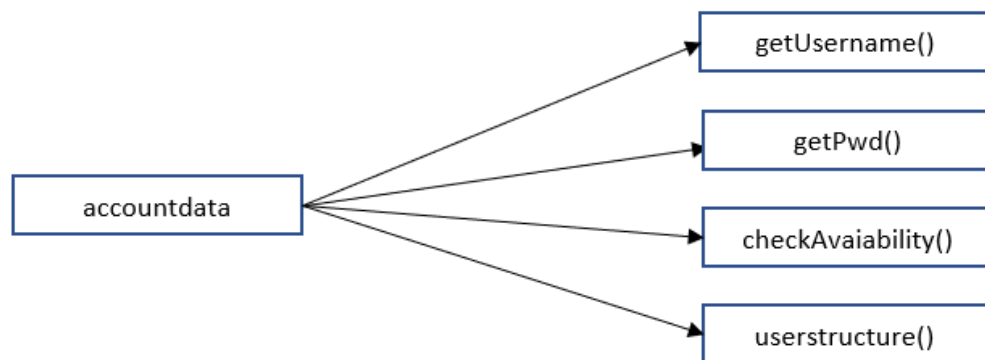


Figure 11: MPC (accountdata)

chatHistory: 0

objectread: 4

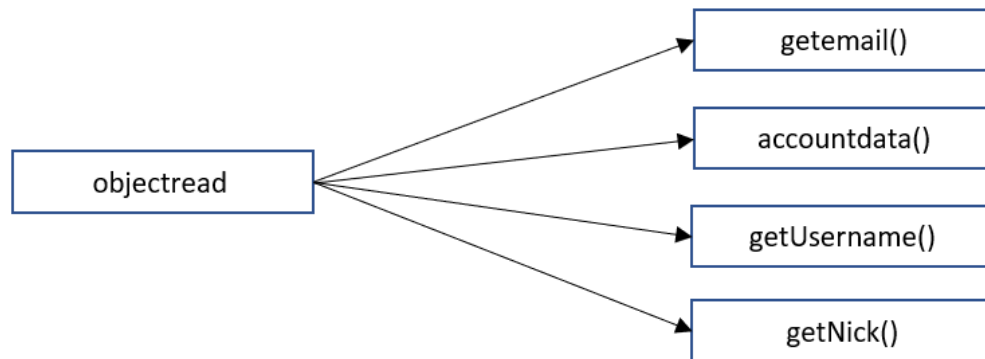


Figure 12: MPC (*objectread*)

createaccountWindow: 3

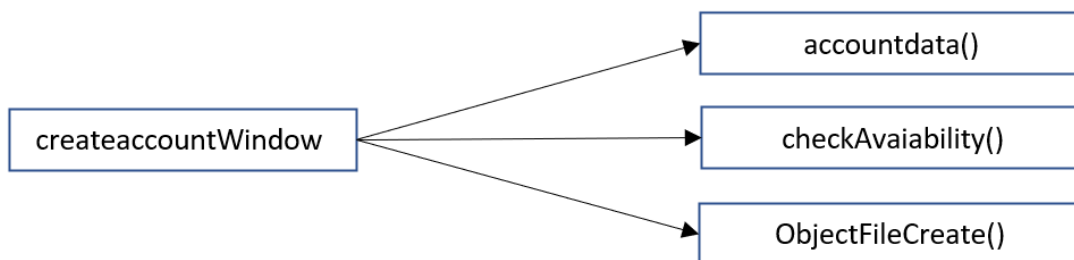
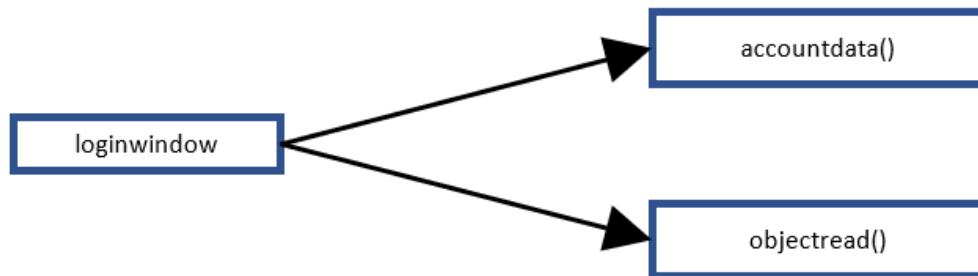


Figure 13: MPC (*createaccountWindow*)



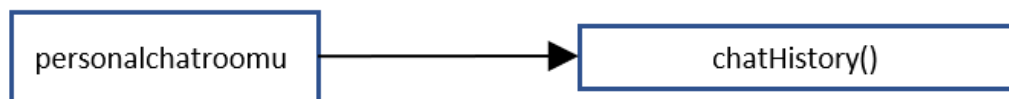
FileShow: 0

loginwindow: 2



*Figure 14: MPC (loginwindow)*

personalchatroomui: 1



*Figure 15: MPC (personalchatroomui)*

ObjectFileCreate: 0

DeleteListedWord: 0

Dictionary: 0

DictionaryMainFrame: 0

## 2. Measuring Cohesion in Object-Oriented System

Cohesion in object-oriented programming refers to the degree to which the methods and properties of a class are related to each other and contribute to a single, well-defined purpose. Cohesion metrics measure how well the methods of a class are related to each other. cohesion refers to the degree to which the elements inside a module belong together. We can measure Cohesion in Object-Oriented System by LCOM (Lack of Cohesion Metric)

### Measurement Process:

$$LCOM = 1 - (\text{sum}(MF) / M * F)$$

Where,

- M is the number of methods in class
- F is the number of instance fields in the class.
- MF is the number of methods of the class accessing a particular instance field.
- Sum(MF) is the sum of MF over all instance fields of the class.

Table 2:Measuring Cohesion (LCOM)

Class	LCOM
userstructure	0.123
accountdata	0
chathistory	1
objectread	1
FileShow	0
ObjectFileCreate	0
createaccountWindow	0
loginwindow	0.5
personalchatroomui	0.192308
serverroom	0
TicTacToy	0.333
TictacMainFrame	0.4
clintsocket	1
clintthread	1
file_name_transfer	-1
clinthandler2	0
serversocket	0
AgeCal	0
Converter	0
DeleteListedWord	0
Dictionary	0
DictionaryMainFrame	-1
DisplayWord	0

ImplementWord	0
WordSearch	0

### 3. Object-Oriented Length Measure

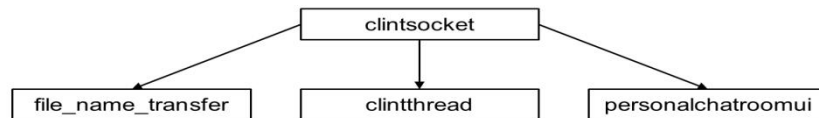
In general, length measurements show the separation between two elements. Distances in object-oriented systems depend on the viewpoint and the model used to describe the proper view of the system. We can gauge it using DIT (Depth Inheritance Tree)

**Measurement Process:** It is done by looking at inheritance tree by looking which class extends or implements which class. We calculate it by OpenStaticAnalyze tool.

*Table 3: Object Oriented Length Measure*

Class	DIT
userstructure	0
accountdata	0
chathistory	0
objectread	1
FileShow	0
ObjectFileCreate	0
createaccountWindow	0
loginwindow	0
personalchatroomui	1
serverroom	0
TicTacToy	0
TictacMainFrame	0
clintsocket	0
clintthread	1
file_name_transfer	1
clinthandler2	0
serversocket	0

AgeCal	0
Converter	0
DeleteListedWord	0
Dictionary	0
DictionaryMainFrame	0
DisplayWord	0
ImplementWord	0
WordSearch	0



25

Figure 16: Depth of Inheritance

## 4. Object-Oriented Reuse Measurement

We can measure Object-Oriented reuse measurement by NOC (Number of Children).

NOC is computed by counting the number of immediate successors (subclasses or subinterfaces) of a class or interface.

**Measurement Process:** Here we count the number of children of each class by looking at the Inheritance tree.

Table 4: Number of Children

Class	NOC
userstructure	0
accountdata	0
chathistory	0
objectread	1
FileShow	0
ObjectFileCreate	0
createaccountWindow	0
loginwindow	0
personalchatroomui	0
serverroom	0
TicTacToy	0
TictacMainFrame	0
clintsocket	3
clintthread	0
file_name_transfer	0
clinthandler2	0
serversocket	0
AgeCal	0
Converter	0
DeleteListedWord	0
Dictionary	0
DictionaryMainFrame	0
DisplayWord	0

ImplementWord	0
WordSearch	0