



## Web Based Tool for Measuring Coupling in Object-Oriented Software Modules

**Vijaykumar Bidve<sup>1\*</sup>****Pakiriswamy Sarasu<sup>2</sup>****Shafi Pathan<sup>3</sup>****Ganesh Pakle<sup>4</sup>**<sup>1</sup>*Marathwada Mitra Mandal's, College of Engineering, Pune, India*<sup>2</sup>*Veltech Dr. RR & Dr. SR Technical University, Avadi, Chennai, India*<sup>3</sup>*Smt. Kashibai Navale College of Engineering, Pune, India*<sup>4</sup>*Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, India*

\* Corresponding author's Email: vijay.bidve@gmail.com

---

**Abstract:** Coupling measurement in object-oriented software is becoming an important area day by day from the software quality evaluation point of view. Coupling is an interdependence relationship between classes (modules) of object-oriented software. The coupling measurement helps to maintain dependency degree between modules of object-oriented software. The moderate coupling values result in quality software. Ultimately quality software retains moderate range of values of software quality attributes such as complexity, efficiency, reusability, maintainability, understandability, flexibility, portability, interoperability, etc. This paper proposes a web based tool for measuring coupling in object-oriented Java software. The idea of a web based tool is to deploy a software coupling tool on intranet or cloud to give access to the prescribed users. The tool can be used online to compute coupling and the resulting data of the tool can be sent/received via the public network. The detailed architecture and components of the tool are described in the paper. The tool is most secured for input processing. The Java and Android projects are evaluated using the tool, the coupling values of Java and Android projects are compared in the results and discussion section of the paper. The Advantages of using the web based tool are also described in the paper. Seven coupling metrics are used from the literature to compute their values using web based tool proposed in this paper. Percentage coupling values of seven coupling metrics are computed for Java and Android projects. The percentage range of coupling values computed in the paper using web based tool is compared with standard range of coupling values described in the literature. The results obtained using web based tool gives us coupling values of Java and Android projects. The coupling values obtained using web based tool proposed in the paper are compared with standard coupling values described in the literature. It is found that the values obtained using the web based tool, are within the standard range of coupling values described in the literature. It means the web based tool proposed in this paper calculates correct coupling values of any object-oriented Java code.

**Keywords:** Coupling, Object-oriented, Quality, Software, Metrics, Measurement, Web.

---

### 1. Introduction

A Coupling is an interdependence relationship between modules of object-oriented software. Coupling measurement helps in software quality control by maintaining range of software coupling values. The moderate range of coupling values leads to quality software. The different software quality attributes like complexity, efficiency, reusability, maintainability, understandability, flexibility, portability, interoperability, etc. can be controlled by keeping a moderate range of coupling values.

To compute coupling values correctly a reliable tool is required. So this paper proposing a tool to measure coupling values amongst the modules of object-oriented software. The additional benefit of this tool is that, it can be deployed on web; hence this tool called as web based tool. It was needed to develop a tool as there was no any tool available in the literature which considers all seven coupling aspects described in table 1. In literature [2, 3, 4] few tools are available like ICAT tool described by the author Jeff Offutt et al. [1] which consider only limited coupling aspects and the tools are not

deployable on web. Now in the era of internet the things must be available on internet hence this tool is having more importance. The novelty of the tool is that, it is a web based tool. Web based coupling measurement tool is not available in the literature so far.

The idea of web based tool is to deploy the coupling measurement tool on intranet or cloud to give access to all the stakeholders. Through the web, it is easy to access the Virtual Private Network (VPN), and the users can send/receive the coupling data on the public network. The access of the tool will only be limited for authorized users. The web based coupling tool is secured as it accepts only zip files as an input. The web based tool takes Java and Android projects as an input for coupling computation. In this work seven coupling metrics are used for coupling measurement using web based tool including parameter coupling, inheritance coupling, global coupling, data abstraction coupling, import coupling export coupling, and external coupling. These seven coupling metrics are defined by the authors of this paper as a part of their previous study [5]. The values of seven coupling metrics are computed using web based tool proposed in this paper and percentage range of each coupling metric type is computed and shown at the end.

The strong points of this tool are- it covers more coupling aspects through seven coupling metrics defined in table 1. In literature no any tool available which could cover as many aspects as web based tool proposed in this paper. Table 1 clearly shows mapping between coupling measures described in the literature and coupling measured used by the authors for development of web based tool. Second strong point is, the web based tool can compute coupling values of Java as well as Android projects. The web based tool is secured as it provides authentication to users and it accepts only .zip files as an input which are less prone to errors. Most important point about the tool is that it is deployable on web and any authentic user can access the tool from any place. Also, the tool uses concept of threads hence the file processing is faster in the tool i.e. in minimum time the tool computes output coupling values.

The rest of paper is organized as follows. Section 2 presents the design of the web based tool for coupling measurement in details. Section 3 discusses the advantages of the web based tool for coupling measurement. Section 4 describes the coupling metrics used in the web based tool for coupling measurement. The results and discussion

are given in section 5. Section 6 confers the conclusion and future.

## 2. Web based tool for coupling measurement

This section presents a web based tool for coupling measurement developed by the authors. The tool is a static Java source code analysis tool; it measures coupling by analyzing Java code statically. The tool accepts a .zip/.java source file as an input. The input Java code's run-time traces (dynamic execution) not required for this analysis. The web based tool for coupling measurement is designed and developed with Java. The architecture diagram of the tool is depicted in Fig. 1. The tool accepts input files called application files as shown in the architecture diagram. As mentioned earlier the tool is restricted to accept only .zip/.java source files as an input.

The middle layer of the architecture constitutes Java platform debug architecture which has three modules zip file extractor, Java application process, number of classes and coupling types respectively. The zip file extractor unzips the input files and retrieves .java files from the folder structure. One by one the files are taken into BufferedReader of Java Virtual Machine (JVM) for further processing. In the processing, extracted files are parsed, and tokens are separated in the form of a class name, method name, attributes, etc. The tool does tokenization using JVM by itself instead of using any third party framework; this is a plus point of the tool. The buffer reader considers one file at a time to parse. It stores the declarations (variables, methods etc.) of the object (class) and maintains the object count. The object count is the table containing count of each object declaration. The count generated is stored in a database table specified by their class names; this task is performed by the module named as number of classes and coupling types. The implementation of the tool uses threads (part of a process) for efficient file processing.

The last layer of the tool includes three modules log generator, coupling generator, graph generator respectively. The log generator maintains a record of the number of objects and data types which are already considered, and/or yet to consider, for coupling computation. The log generator module obtains data from the module Java application process. Using the tokenized objects and coupling metrics described in section 4 of the paper, values of coupling metrics are computed and stored in the database. The coupling generator retrieves the

Table 1. Coupling metrics used in web based tool

Sr. No.	Coupling Metrics Used	Coupling Mechanism	Related Coupling Metrics in the Literature
1	Parameter coupling	Method of one class invokes method/passes parameter/passes message to methods of other classes or to make a call to the constructor of other class.	CBO [7, 8]
			Interaction Coupling [9]
			MM [10, 11]
			IC_OM, IC_CM, EC_OM, EC_CM [12]
			Invocation Coupling [13]
2	Inheritance coupling	One class is a super-class of another class (Inheritance).	MI [14]
			CBO, RFC [8]
3	Global coupling	Method of one class can directly access parts of the internal structure, of another class's method (friend). Also to access common, shared, non-local variables of other classes.	Inheritance Coupling [9, 13, 14]
			Interaction Coupling [9]
			Friend Coupling [10, 11]
4	Data abstraction coupling	One class is used in the implementation of methods of other class. One class is the domain of the instance variable, the local variable of other class.	AR [14]
			Component Coupling [9]
			CM, CA [11]
5	Import coupling	All type of coupling due to any import mechanism.	MP, MR, DA [14]
			Import Coupling [11, 14, 15, 16]
6	Export coupling	All type of coupling due to an export mechanism.	Export Coupling [11, 14, 17, 18]
7	External/file coupling	Sharing an external device like the printer, HDD, external file by the two classes.	External Coupling [1, 9, 19, 20, 21, 22]

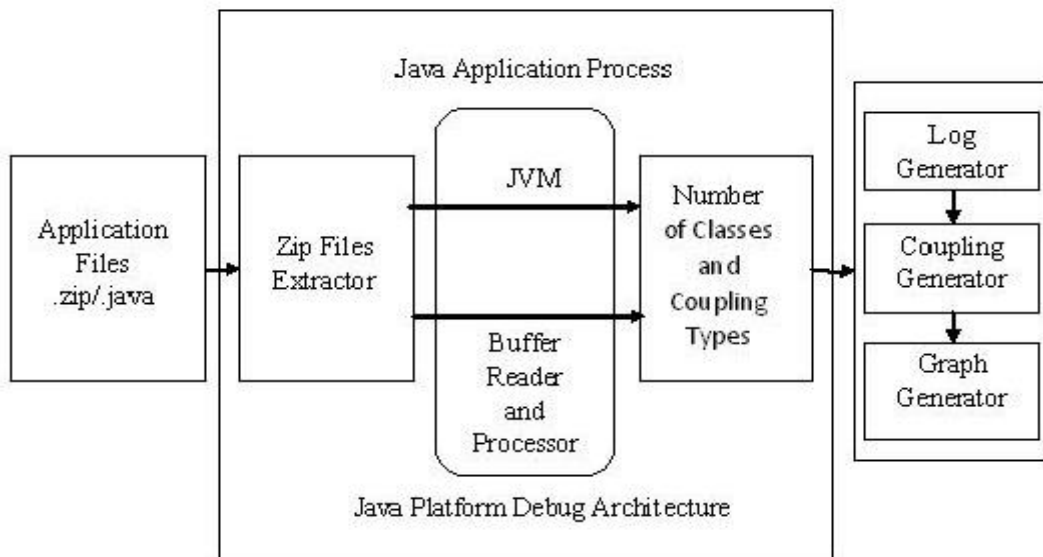


Figure.1 Architecture diagram of web based tool for coupling measurement

respective coupling values from the database and is displayed on the dashboard using Java Server Pages (JSP) files. The graph generator module generates the graphs of coupling metrics values. This module can generate three types of graphs involving pie chart, bar chart, and line chart respectively. The graphs are helpful to compare values of different coupling metric types.

The operating steps of the web based tool are very simple and given subsequently. As the tool is a web based application, registered users can have the

access of the tool. The new users can also do the registration by submitting required details. After successful login, the security key is provided to the users on his/her mobile as well as email address. The user is authenticated with the security key for the further access of the tool. After the successful login and authentication, the user is allowed to evaluate his/her input Java projects. The user can browse required project location to select input Java project. The tool maintains a log of each produced output with the options to view coupling values, pie

graph, line chart, and bar chart. The view coupling option provided in the dashboard of the tool displays class-wise coupling values of each Java project. Similarly, the pie graph, line graph, and bar graph shows the summing coupling values of all the classes of input project for each coupling metric defined in Table 1.

Fig. 2 shows the screenshots of the various operational steps of the web based tool. User can give any Java or android project as an input to the web based tool. User can easily browse and select any input file from the computer system. The tool processes input file and produces class wise coupling values of project as an output. The tool maintains log of all earlier evaluated projects. Using the log maintained by the tool, users can anytime view coupling values of all earlier evaluated projects. No need to evaluate the project more than once; once the project is evaluated its coupling values will be stored in system log and can be viewed in future.

### 3. Advantages of web based tool for coupling measurement

The web based tool for coupling measurement has following advantages,

- i. **Web application:** As this a web based application, it can be deployed on intranet or cloud to give access to all stakeholders. This makes easy, to access the website through Virtual Private Network (VPN). The data can be sent/received on the public network.
- ii. **Secure input files:** The zip files are given as an input to the tool for security reasons, so HTTP's protocol and port 80 does not accept .exe, .jar, and NTFS files. Generally, viruses get append themselves to executable files to get loaded into memory and spread. Archive files like .zip usually do not get infected as they are not executable. Hence, web based tool is secured for accepting user input files.
- iii. **Rapid file processing:** The input file processing and classes finding tasks are divided into threads to make the scanning of input file more efficient and express.
- iv. **.java input:** As this tool is developed in Java the source of the tool can be used as an input to compute coupling metric values. The resulting output coupling values can be compared with the standard range of coupling values.
- v. **Android projects as an input:** This tool also accepts android projects as an input and produces results for them. The tool skips all

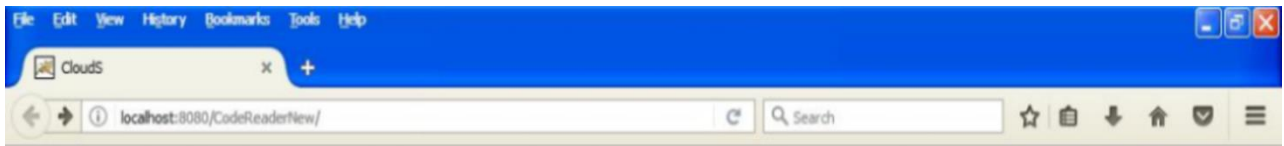
other manifest files, XML files so its native architecture does not be affected.

### 4. Coupling metrics used

Selection of coupling metrics is a crucial task in software coupling measurement. There are multiple coupling metrics described and made available in the literature by various authors. Every author have described his metrics with distinct measures and coupling mechanisms. There is no unique standard metrics available in the literature for coupling measurement. The many metrics described in the literature are not covering all aspects of coupling, few metrics are ambiguous. The authors of the paper have identified this gap and worked on describing a normalized coupling metrics. Authors surveyed multiple papers on coupling measurement metrics and worked on it. The various coupling metrics with limited coupling mechanisms, overlapped coupling mechanisms, and ambiguous coupling mechanisms are identified by the authors from the literature. The authors of this paper have described a new coupling metrics with seven measures as given in Table 1.

The newly defined metrics is mainly comprehensive i.e. covers all aspects of coupling. There is no overlapping from coupling mechanisms point of view. Every coupling mechanism constitutes to different coupling aspect and all coupling mechanisms are covered in the newly defined metrics by the authors given in Table 1.

The web based tool has used a normalized coupling metrics defined in the literature [5, 6] by the authors. As told earlier this coupling metrics are defined by the authors of this paper as a part of their previous study. The normalized metrics covers all aspects of coupling and is without overlapping in terms of its coupling mechanisms. The main thing about the metrics is that, it covers almost all aspects of coupling described in the literature. Table 1 presents the coupling metrics used by the web based tool for computing coupling values. The last column of Table 1 relates the coupling metric described in the literature by various authors with newly defined coupling metrics by the authors of this paper. The coupling metrics proposed in column 2 of the Table 1 are applicable to all object-oriented programming languages.




## Code Reader

Home User

### Client Login

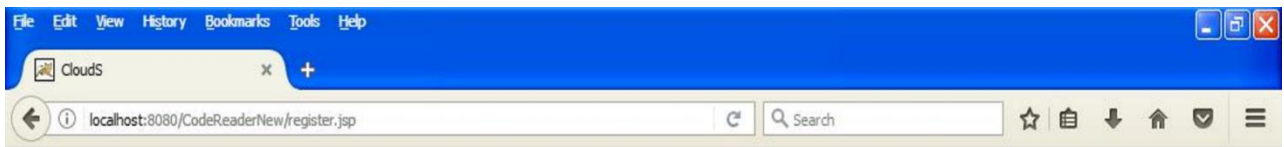
User id:

Password:



Code Reader

(a)



## Code Reader

Home User

### Client Registration

Name  Last Name

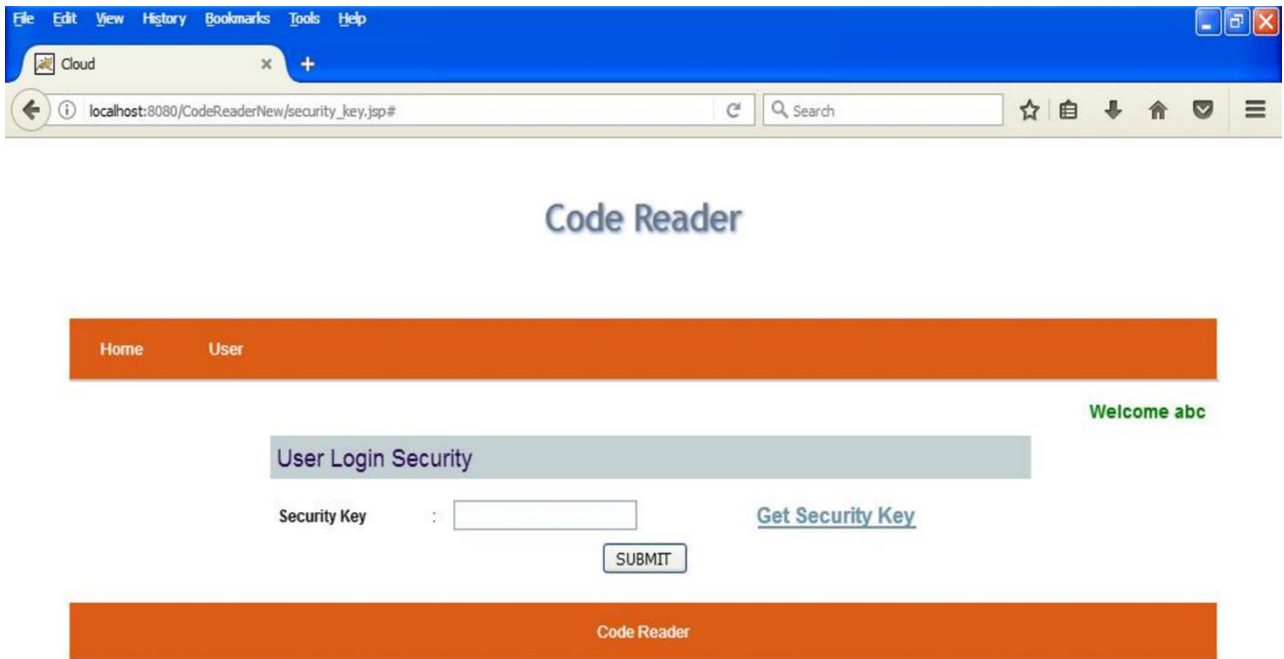
Age  Gender  Male  Female

Email

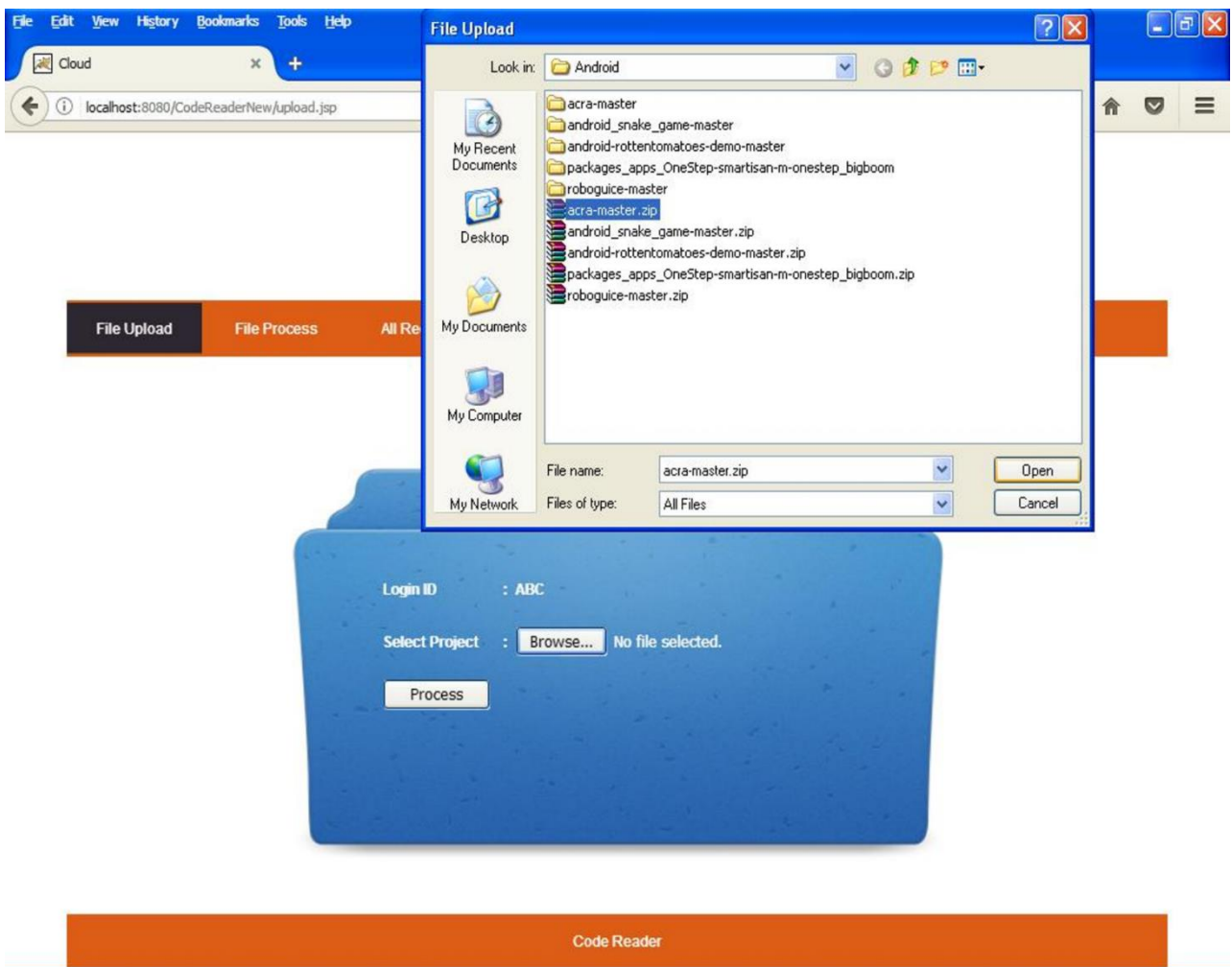
Username  Password

Code Reader

(b)

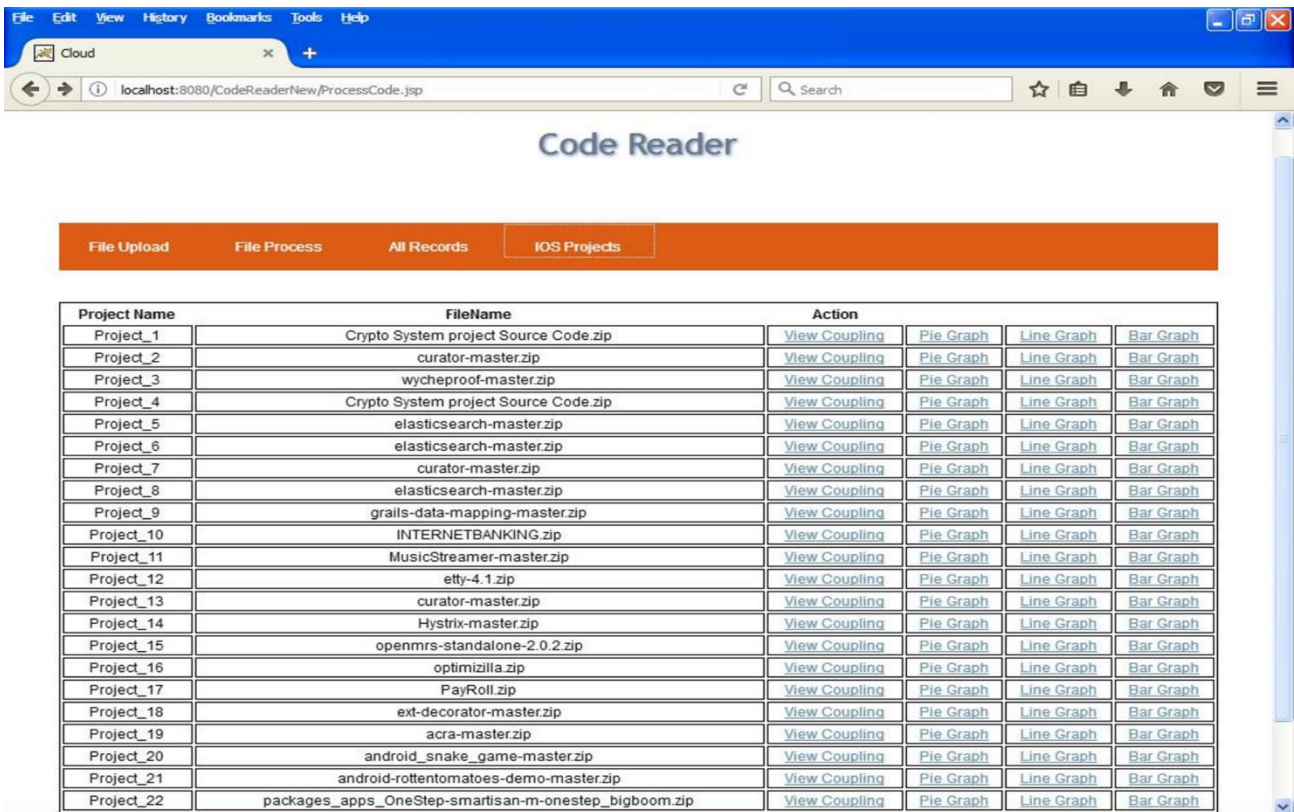


(c)

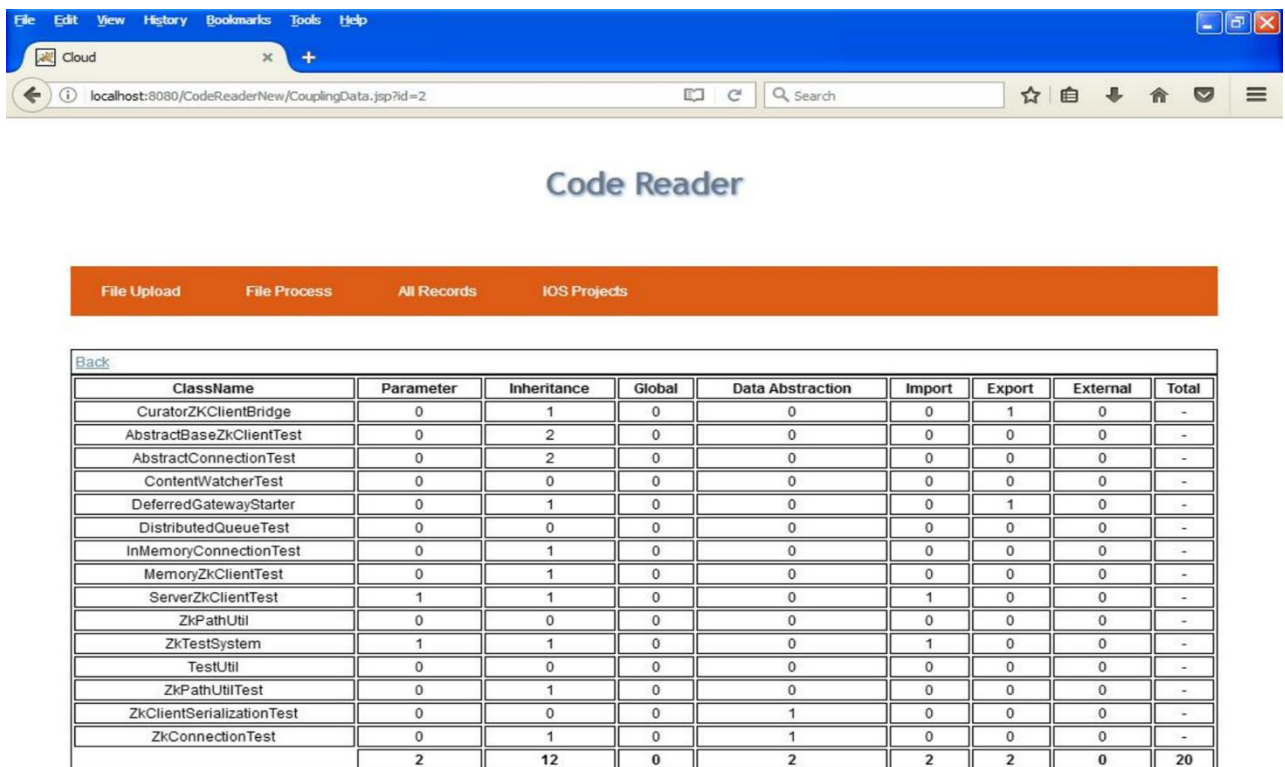


(d)





(e)



(f)

Figure.2 Screen shots of web based tool (Operational steps): (a) login window, (b) registration window, (c) security key generation window, (d) input browsing window, (e) log of produced outputs, and (f) class wise coupling values of individual project

Table 2. Coupling values of Java projects obtained from web based tool

Coupling Metrics-> /Java Application	Parameter coupling	Inheritance coupling	Global coupling	Data abstraction coupling	Import Coupling	Export Coupling	External/file coupling
Curator-master	2	12	0	2	2	2	0
PayRoll	8	18	0	19	8	8	0
Crypto System Project Source Code	40	14	1	10	41	41	3
Grails-data-mapping-master	1124	598	16	742	639	687	0
Hystrix-master	1637	372	401	727	1465	1463	0
MusicStreamer-master	2204	1168	308	804	2090	2128	7
Total	5015	2182	726	2304	4245	4329	10
Average	836	364	121	384	708	722	2

Table 3. Coupling values of Android projects obtained from web based tool

Coupling Metrics-> /Android Application	Parameter coupling	Inheritance coupling	Global coupling	Data abstraction coupling	Import Coupling	Export Coupling	External/file coupling
Acra-master	243	101	0	138	213	212	10
Snake_game-master	4	5	0	3	4	4	0
Rottentomatoes-demo-master	3	4	0	4	2	2	0
Packages_apps	509	117	230	269	611	611	4
Total	759	227	230	414	830	829	14
Average	190	57	58	104	208	207	4

## 5. Results and discussion

As a result, the web based tool evaluates Java/Android projects to compute values of seven coupling metrics described in section 4. In this paper, six Java projects and four Android projects are evaluated, and coupling values are obtained. All the input Java projects are taken from the repository Sourceforge.net. Table 2 shows the resulting coupling values of input Java projects obtained from

the web based tool. The last two rows of Table 2 show sum and average values respectively of each type of coupling metric. The average value of each coupling type is calculated to find out the percentage value of each coupling type among the projects. The pie chart of the percentage coupling values of each coupling type for Java projects is shown in Fig. 3. The range of coupling values of Java projects obtained in the Table 2 is compared with the



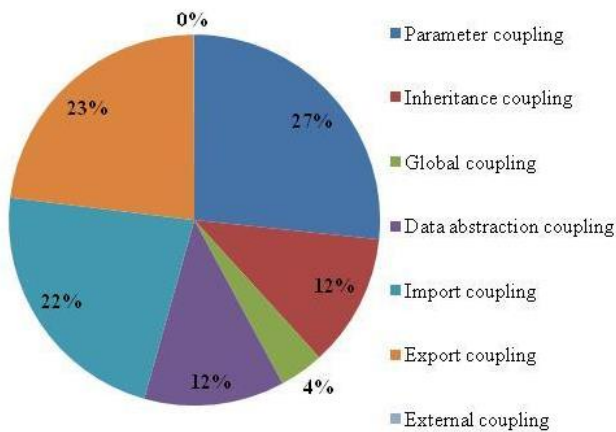


Figure.3 Percentage value of each coupling type in Java projects

Table 4. Threshold coupling values [6]

Sr. No.	Coupling Type	Threshold Coupling % range
1	Parameter Coupling	24 to 34
2	Inheritance Coupling	3 to 13
3	Global Coupling	0 to 7
4	Data Abstraction Coupling	9 to 19
5	Import Coupling	18 to 28
6	Export Coupling	19 to 29
7	External Coupling	0 to 5

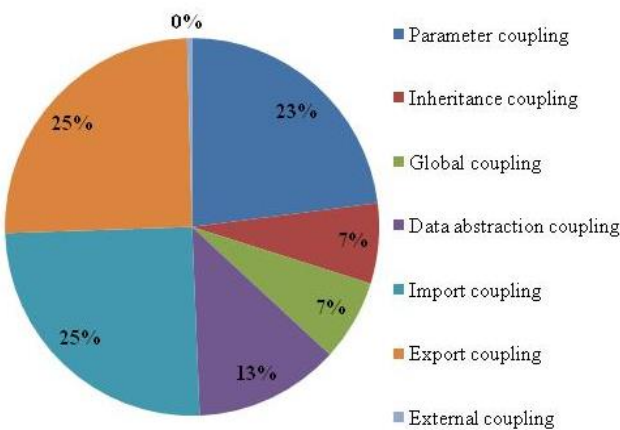


Figure.4 Percentage value of each coupling type in Android projects

standard range of coupling values defined in the literature as given in Table 4. It is observed from Fig. 3 that, the percentage range of coupling values exactly lies in between the threshold range defined by the authors V. S. Bidve et al. in the literature [5, 23].

Hence the results of coupling values obtained using the web based tool of Java projects are in the standard threshold range defined in the literature by authors V. S. Bidve et al. [5, 23]. Hence authors of this paper are coming to the conclusion that, the web based tool described in the paper gives correct coupling values of Java projects.

Similarly, Table 3 shows the resulting coupling values of Android projects obtained from the web based tool. The last two rows of Table 3 shows the sum and average values respectively of each coupling type for four Android projects. The pie chart of the percentage coupling values of each coupling type for Android projects is shown in Fig. 4. The range of coupling values obtained from web based tool is again compared with the standard range of coupling values given in Table 4. It is observed from Fig. 4 that, the percentage range of coupling values exactly lies in between the threshold range defined by V. S. Bidve et al., except for parameter coupling. In case of parameter coupling the values obtained for Android projects are slightly small than threshold values given in Table 4. There is no major difference in the values of parameter coupling. The parameter coupling values obtained in this work for Android projects is near to lower value parameter coupling described in the literature as given in Table 4. So, this again ensures that the web based tool computes correct coupling values for all type of coupling measures.

At the end, the results obtained from web based tool and the results available in the literature as given in Table 4 leads to the conclusion that, the web based tool correctly calculates values of seven coupling metrics defined in the Table 1. These resulting coupling values are helpful for software developers to maintain level of coupling amongst various modules of object-oriented software. If the coupling values of software are greater than the range of values given Table 4 then the software's design will be considered as more complex design. If the coupling values are within the range as given in Table 4 then the software will be considered as well designed software from coupling point of view. Otherwise if the coupling values of software are smaller than the range given in Table 4 then the design will be considered as fewer complexes but reusability component of code from other modules will be less. Hence coupling values of software must be moderate as given in the Table 4. Using web based tool standard projects from the repository Sourceforge.net are evaluated. The projects are well designed as there coupling values are within the threshold range given in Table 4. This further leads to the conclusion that the web based tool computes

correct coupling values and this tool is a guide for developer to maintain coupling values within range. If coupling values exceeds their limits those can be controlled by changing design of software.

## 6. Conclusion and future work

This paper presents a web based tool for measuring coupling in object-oriented software modules. The software professionals are the beneficiaries of the tool. The coupling values obtained using this tool are within the threshold range of coupling [5, 23]. The tool helps to software developers to maintain coupling values in the standard range given in the literature and described in Table 4. The tool accepts Java and Android projects as an input and computes values of coupling metrics. The tool is useful for online coupling computations and to share results on the public network. The tool can be accessed online via virtual private network (VPN). The authorization is required to access the web based tool. The tool accepts input in the form of .zip files to reduce virus attack. The tool is developed with the inclusion of thread concept; hence the input file processing is fast. As per results the coupling values obtained using the tool are in the standard range of coupling values as per literature [5]. Hence the obtained results leads to the conclusion that, the coupling values obtained using web based tool are correct. Also, it is observed that, Java projects have more interactions in the form of method-method coupling i.e. parameter coupling is more; it is good sign of reusability. Whereas Android projects have comparatively more interactions in the form of external coupling i.e. use of global variable is more; it is not a good programming practice. In future, the tool can be extended to work for other object-oriented programming languages like .Net, C++ etc.

## References

- [1] J. Offutt, A. Abdurazik, and S. Schach, "Quantitatively Measuring Object-Oriented Couplings", *Springer's Software Quality Journal*, Vol. 6, No. 4, pp.489-517, 2008.
- [2] Zakrani, M. Hain, and A. Namir, "Software Development Effort Estimation Using Random Forests: An Empirical Study and Evaluation", *International Journal of Intelligent Engineering and Systems*, Vol.11, No.6, pp. 300-311, 2018.
- [3] M. Bansal and C. Agrawal, "Jaya Algorithm and Artificial Neural Network Based Approach for Object-Oriented Software Quality Analysis", *International Journal of Intelligent Engineering and Systems*, Vol.11, No.4, pp. 275-282, 2018.
- [4] M. Gopal, "Design Quality Metrics on the Package Maintainability and Reliability of Open Source Software", *International Journal of Intelligent Engineering and Systems*, Vol.9, No.4, pp. 195-204, 2016.
- [5] V. Bidve and P. Sarasu, "Comprehensive Survey of Coupling Metrics in Object-Oriented Software", *International Journal of Applied Engineering Research*, Vol. 10, No. 14, pp.34687-34698, March 2015.
- [6] V. Bidve and P. Sarasu, "Tool for Measuring Coupling in Object-Oriented Java Software", *International Journal of Engineering and Technology*, Vol. 8, No. 2, pp.812-820, 2016.
- [7] S. Chidamber and C. Kemerer, "Towards a Metrics Suite for Object-Oriented design", In: *Proc. Conference on Object-Oriented Programming: Systems, Languages and Applications*, Vol. 26, No. 11, pp.197-211, 1991.
- [8] S. Chidamber and C. Kemerer, "A Metrics Suite for Object-Oriented Design", *IEEE Transactions on Software Engineering*, Vol. 20 No. 6, pp.476-493, 1994.
- [9] J. Eder, G. Kappel, and M. Schrefl, "Coupling and Cohesion in Object-Oriented Systems", *Technical Report, University of Klagenfurt*, 1994.
- [10] L. Briand, P. Devanbu, and W. Melo, "An Investigation into Coupling Measures for C++", In: *Proc. of the 19th Int', International Conf. Software Eng.*, pp.412-421, 1997.
- [11] Lionel, W. John Daly, and K. Jurgen, "A unified framework for coupling measurement in object-oriented systems", *Software Engineering, IEEE Transactions* Vol. 25, No.1, pp. 91-121, 1999.
- [12] E. Arisholm, C. Lionel, and A. Føyen, "Dynamic Coupling Measurement for Object-Oriented Software", *IEEE Transactions on Software Engineering*, Vol. 30, No. 8, 2004.
- [13] H. Li, "A Novel Coupling Metric for Object-Oriented Software Systems", *IEEE International Symposium on International Journal of Computer Applications*, Vol. 27, No.10, pp. 609-612, 2008.
- [14] H. Sukainah and O. Alan, "A Coupling and Cohesion Metrics Suite for Object-Oriented Software", In: *Proc. of International Conference on Computer Technology and Development*, Vol.1, No.3, pp.421-425, 2009.
- [15] M. English, T. Cahill, and J. Buckley, "Construct specific coupling measurement for

- C++ software”, *Computer Languages, Systems & Structure*, Vol. 38, No. 2, pp.300–319, 2012.
- [16] S. Yeresime, J. Pati, and S. Rath, “Review of Software Quality Metrics for Object-Oriented Methodology”, In: *Proc. of International Conference on Internet Computing and Information Communications, Advances in Intelligent Systems and Computing*, pp. 267-278, 2014.
- [17] A. Mitchell and J. Power, “Toward a definition of run-time object-oriented metrics”, In: *Proc. of the Seventh ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering*, 2003.
- [18] A. Mitchell and J. Power, “Runtime Coupling Metrics for the Analysis of Java Programs”, *Preliminary Results from SPEC and Grand Suites*, 2003.
- [19] A. Mitchell and J. Power, “An empirical investigation into the dimensions of run-time coupling in java programs”, In: *Proc. of Third Conference on the Principles and Practice of Programming in Java*, pp.9-14, 2004.
- [20] Mitchell and J. Power, "Using object-level run-time metrics to study coupling between objects", In: *Proc. of the 2005 ACM Symposium on Applied Computing*, 2005.
- [21] J. Zhao, "Measuring Coupling in Aspect-Oriented Systems", In: *Proc. of the 10th IEEE International Soft. Metrics Symposium*, 2004.
- [22] W. Li and S. Henry, "Object-oriented metrics that predict maintainability", *Journal of Systems and Software*, Vol. 23, No. 2, pp.111-122, 1993.
- [23] V. Bidve and P. Sarasu, “Coupling Measures and its Impact on Object-Oriented Software Quality”, *Journal of Science and Technology*, Vol. 9, No. 21, pp.1-7, 2016.