SPL-1 Project Report,[2018]

Code Metric Analysis

Course : SE – 305 course name: Software Project Lab -1

Submitted by:
Amran Hossain
BSSE Roll No: 0917
Session : 2016-17

Supervised by:
Md. Saeed Siddik
Lecturer
Institute of Information Technology

Institute of Information Technology
University of Dhaka.
Date: 30-05-18

Table of contents

Table of figures

1. Introduction:

This project analyze about the code metrics .Code metrics can be produced by static code analysis tools to determine complexity and non-standard practices. Here analysis the code metrics , which is covering a huge part in computer science . I have completed some of task of code metrics . These are given below,

* lines of code

* number of variables

* blank lines of code

* number of comment lines

* number of methods

* number of parameters

1.1 Importance of Code metrics:

Code metrics help the developers/programmers to find out the problem base code that may need to refactor the code.

1.2 Broad Domain:

The project was developed in c++ platform . For this metrics is quantitative measure of the degree to which a system, component, or process possesses a given attribute. [1] code metrics is such kind of form. It has to be analyzed by any software measurement tools.

1.3 Background of study:

To complete this project I have done it in raw coding. As I had no idea before this project ,it was totally new . So I needed more to learn about c++ language. About the built in function which are needed in the code I had to be known about it. Such as to read from file <fstream> header file to be used and written ifstream for input from file .

```
#include <fstream>
………………
……………...
```

```
ifstream ifile; // ifile is the file name
```

After read file I have to firstly count total lines of code/ how many lines in a code. Next I have counted how many variables in a code . Variables are used to store information to be referenced and manipulated in a computer program.[3] V. Hence variables hold data in programming language .As I have to know about which variables are primitive type and which are the object type . For this primitive type declared by default . Such as int,char,string,void,boolean,float,double etc are primitive type of variables . In object type of variables are created by user.

Next I had counted how many comment lines in a code. For this purposes , I had to know details about the comments. Comment is a programmer-readable explanation in the source code of a computer program. They are added with the purpose of making the source code easier for humans to understand. The syntax of comments in various programming languages varies considerably. There are two kinds of comments. 1. Single line comments (//) 2. Multi line comments .(/*… */)

```
/*
void EventLoop();
void EventLoop();
*/

// This comment will also comment the following line
```

fig 1: multi line comment and single line comment [2]

Next I had to be checked method and counted the number of method . For this I have to know first about the method . A method in OOP is a procedure associated with a message and an object . A method has a name and return type. For this purposes I have checked '(' opening bracket in line. Thus I have to check method and lastly I have not count the default method. Like for() ,if() , main () etc.

Next I checked the parameter . The term *parameter* is often used to refer to the variable as found in the function definition. Like,

```
double sales_tax(double price)

{
    return 0.05 * price;
}
```

Here price is a parameter of this method. For this,calling method,s parameter two ways, call by references & call by values .Then parameter checking I have to work method code than checking inside the methods parameter comma separated.

fig-2: snippet code of parameter check.

Lastly coupling and cohesion. These are the new term.So I have read more about it. Coupling is relation among the class/interface and cohesion is relation within the class. Such as,

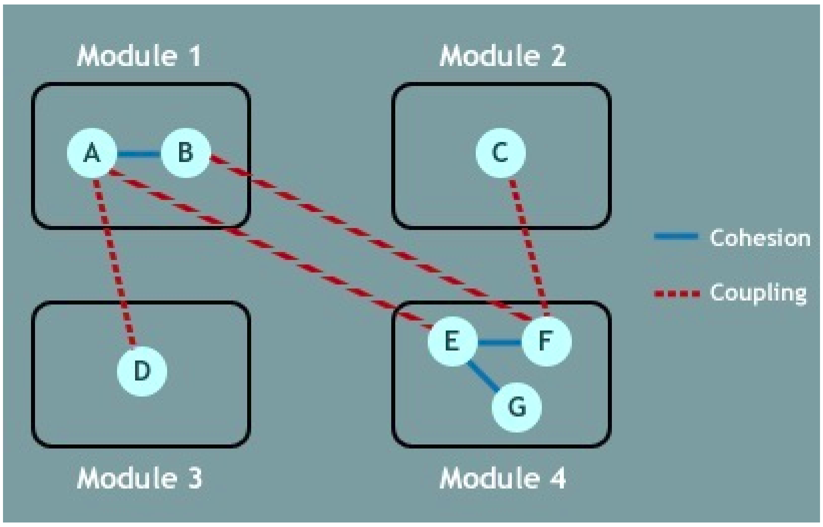


fig - 2: coupling and cohesion module [3]

2.Challenges:

As a beginner my project has some challenging part to do. When I extract variables in object type it is challenging part to do . For object type variables when I got '=' sign than new..() than I encounter this is a variable. Also in method checking some methods are by default . Such like as for () , if() , else.. if(),main (),some instance of objects and others. So I have to check only creating methods by own . Also the main challenging part of this project is coupling and cohesion check from the java file . I had no basic idea about the coupling and cohesion before . I read it than execute my programmes. As I know coupling is the relation between two or more classes and cohesion is only own classes relationship. For this my supervisor helped me much and I have to read more internet sites for known to well about these topics . After knowing overviews from these topics then I have tried to implemented into the code .After a long term ,I had succeded about its.

By the way, this project has helped me improving coding skill. Also thinking skill. I learnt many of new things whatever I didn't know before . So taking many of challenging part is very helpful to me.

3.Project Overview :

Now this time the project task which is given by a diagram below. Here my progessive view has given that.Here timeline graph:

	Month 1	Month 2	Month 3	Month 4	Month 5
Requirement analysis	■	■			
Proposal submission		🟦			
Coding		🟧	🟧	🟧	🟧
Testing					■
Report and powerpoint presentation					🟫

fig -3 : timeline diagram1

Here is overview of my project part by part complete in monthly view,

Tasks\Months	January	February	March	April	May
countLine		✓			
numberOfvariables		✓			
countComment Line number					
countMethd				✓	
countParameter				✓	
coupling					✓
cohesion					✓

fig - 4 : timeline diagram 2

4.User manual:

There are some requirements for developing the project and running the code as far as possible.

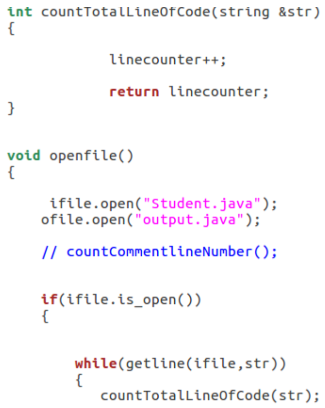
4.1 Software :

The code is implemented in C++ platform . Source code/file is taken on a java file . For this, the code will be run in codeblocks , any text editor for c/c++ platform like (gedit) .

The code run in windows(7,8,10),linux,linux mint, ubuntu (any version) in desktop or laptop.

4.2 Functional manual:

To see the result of task you can see the result in terminal of codeblocks or any text editor terminal . Also can see the result in another java file as a output of the programme . As code is written in c++ platform it's need to g++ compiler. Reading file from line by line and incrementing it than I found the total line of Code. Here is the snippet



```
int countTotalLineOfCode(string &str)
{

            linecounter++;

            return linecounter;
}

void openfile()
{

     ifile.open("Student.java");
    ofile.open("output.java");

    // countCommentlineNumber();


    if(ifile.is_open())
    {


        while(getline(ifile,str))
        {
           countTotalLineOfCode(str);
```

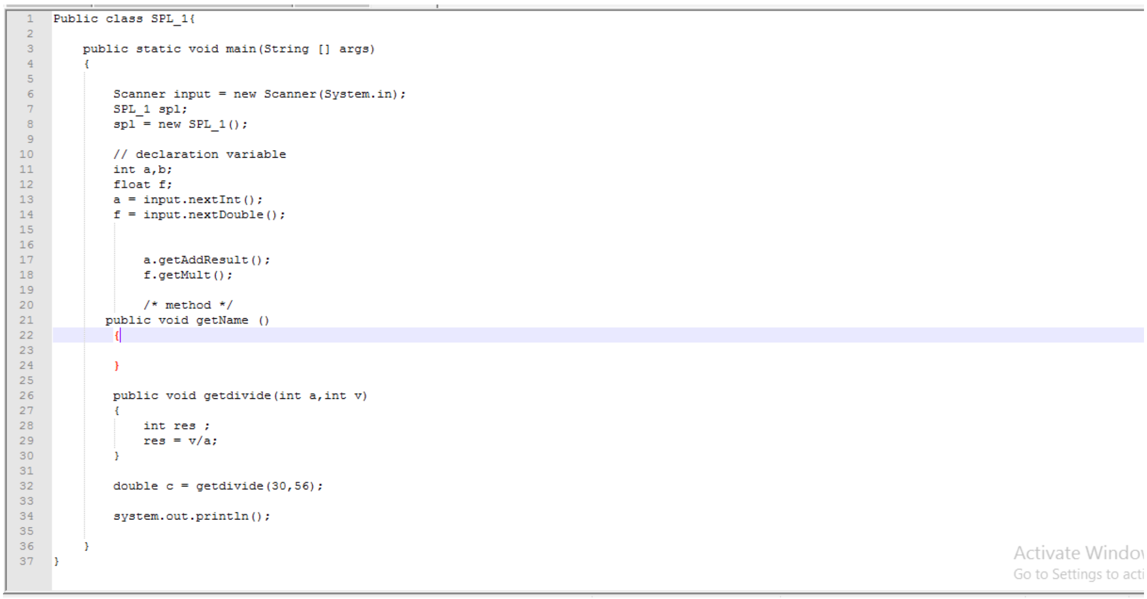
fig -5: snippet of countTotalLine

5.Conclusion:

I have learned a lot through this project. This project has sharpened my coding skill . Also helping me analyzing the code. I learned from this project how can extract code . Yet before, I had no idea about this or any thing of this project. Infact code metrics is a huge part in the code analysis.
Also remain Maintainability Index,Cyclomatic complexity, Depth Inheritance,Number of fields,Type rank, Method rank,Instability and abstractness etc.[4] Next I will try to find out this metrics from the code so that I can complete analyze the whole code metrics. For knowing better and improve myself in coding and thinking skill.

6.Appendix:

Actually I had read a java file means a file which is written in java format. Like this given a sample java file as a input file, which was analyzed and find out this methods,parameters,variables etc.

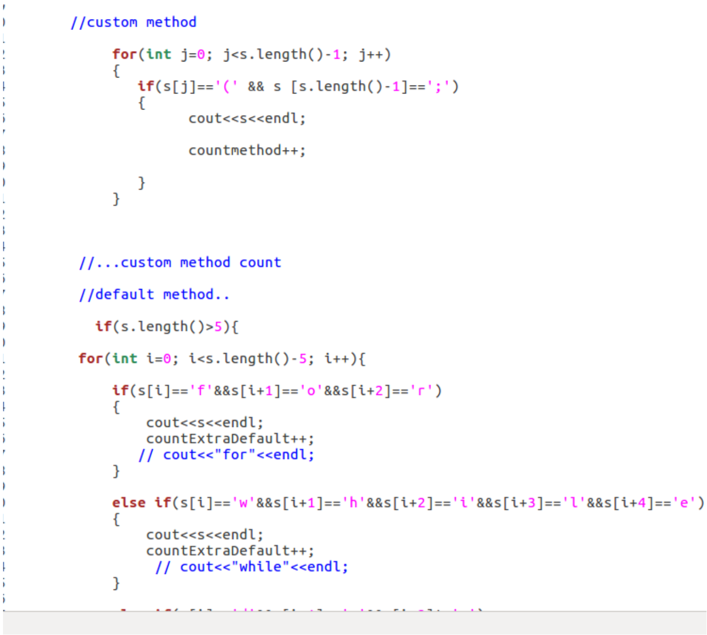


```
Public class SPL_1{

    public static void main(String [] args)
    {

        Scanner input = new Scanner(System.in);
        SPL_1 spl;
        spl = new SPL_1();

        // declaration variable
        int a,b;
        float f;
        a = input.nextInt();
        f = input.nextDouble();


            a.getAddResult();
            f.getMult();

            /* method */
       public void getName ()
        {

        }

        public void getdivide(int a,int v)
        {
            int res ;
            res = v/a;
        }

        double c = getdivide(30,56);

        system.out.println();

    }
}
```

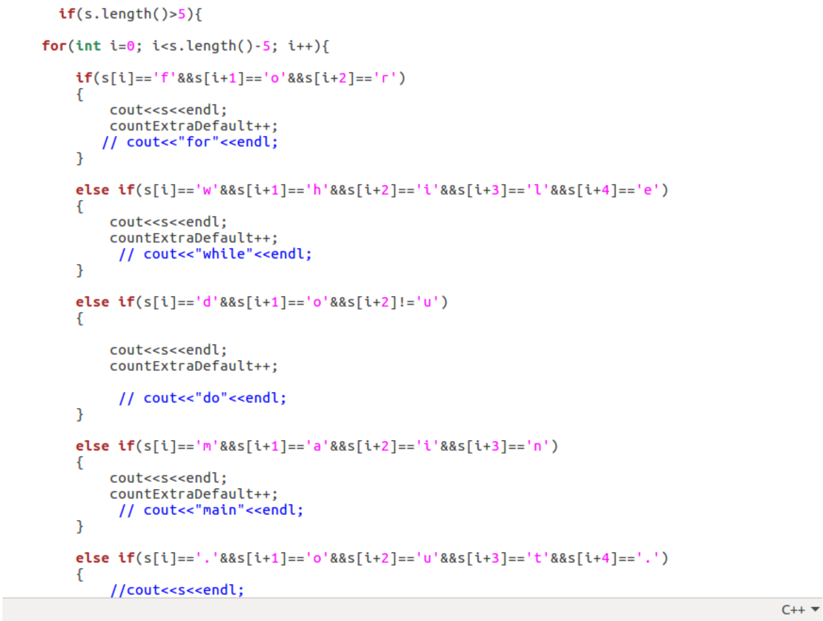
fig - 6: java input file

13

For checking method now I give a snippet of this code portion,



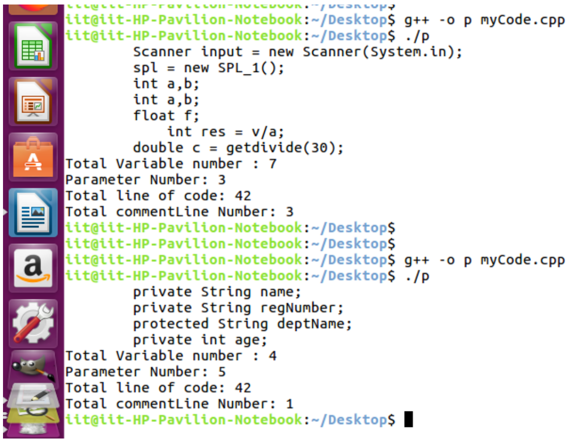
```
//custom method

     for(int j=0; j<s.length()-1; j++)
     {
        if(s[j]=='(' && s [s.length()-1]==';')
        {
              cout<<s<<endl;

              countmethod++;

        }
     }


 //...custom method count

 //default method..

   if(s.length()>5){

 for(int i=0; i<s.length()-5; i++){

     if(s[i]=='f'&&s[i+1]=='o'&&s[i+2]=='r')
     {
         cout<<s<<endl;
         countExtraDefault++;
        // cout<<"for"<<endl;
     }

     else if(s[i]=='w'&&s[i+1]=='h'&&s[i+2]=='i'&&s[i+3]=='l'&&s[i+4]=='e')
     {
         cout<<s<<endl;
         countExtraDefault++;
          // cout<<"while"<<endl;
     }
```



```
  if(s.length()>5){

for(int i=0; i<s.length()-5; i++){

    if(s[i]=='f'&&s[i+1]=='o'&&s[i+2]=='r')
    {
        cout<<s<<endl;
        countExtraDefault++;
       // cout<<"for"<<endl;
    }

    else if(s[i]=='w'&&s[i+1]=='h'&&s[i+2]=='i'&&s[i+3]=='l'&&s[i+4]=='e')
    {
        cout<<s<<endl;
        countExtraDefault++;
         // cout<<"while"<<endl;
    }

    else if(s[i]=='d'&&s[i+1]=='o'&&s[i+2]!='u')
    {

        cout<<s<<endl;
        countExtraDefault++;

         // cout<<"do"<<endl;
    }

    else if(s[i]=='m'&&s[i+1]=='a'&&s[i+2]=='i'&&s[i+3]=='n')
    {
        cout<<s<<endl;
        countExtraDefault++;
         // cout<<"main"<<endl;
    }

    else if(s[i]=='.'&&s[i+1]=='o'&&s[i+2]=='u'&&s[i+3]=='t'&&s[i+4]=='.')
    {
        //cout<<s<<endl;
```

fig -7: snippet of method checking

Now the output is



```
iit@iit-HP-Pavilion-Notebook:~/Desktop$
iit@iit-HP-Pavilion-Notebook:~/Desktop$ g++ -o p myCode.cpp
iit@iit-HP-Pavilion-Notebook:~/Desktop$ ./p
        Scanner input = new Scanner(System.in);
        spl = new SPL_1();
        int a,b;
        int a,b;
        float f;
            int res = v/a;
        double c = getdivide(30);
Total Variable number : 7
Parameter Number: 3
Total line of code: 42
Total commentLine Number: 3
iit@iit-HP-Pavilion-Notebook:~/Desktop$
iit@iit-HP-Pavilion-Notebook:~/Desktop$
iit@iit-HP-Pavilion-Notebook:~/Desktop$ g++ -o p myCode.cpp
iit@iit-HP-Pavilion-Notebook:~/Desktop$ ./p
        private String name;
        private String regNumber;
        protected String deptName;
        private int age;
Total Variable number : 4
Parameter Number: 5
Total line of code: 42
Total commentLine Number: 1
iit@iit-HP-Pavilion-Notebook:~/Desktop$ █
```

fig-8 : snippet of output

7.References:

1. **www2.southeastern.edu/Academics/Faculty/galkadi/285/notes/Chapter15.doc** last access 28.05.18

2. **https://ardalis.com/static-code-analysis-and-quality-metrics** last access 11.02.18

3. **https://launchschool.com/books/ruby/read/variables** last access 23.03.18