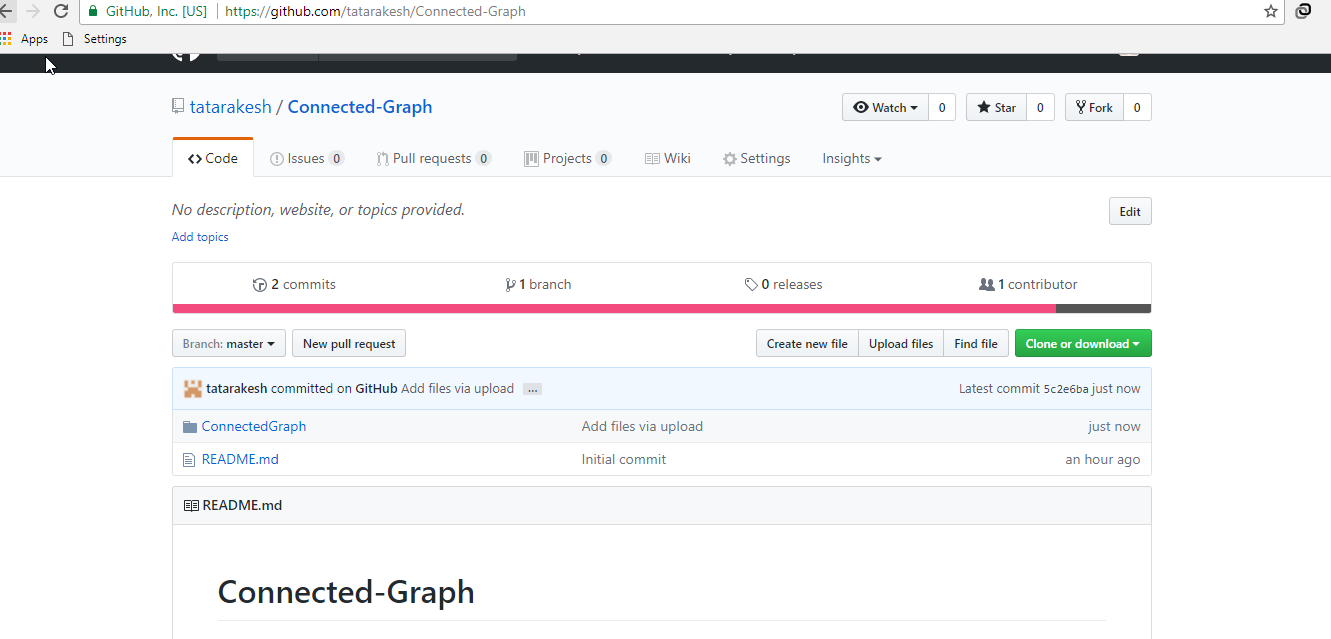


**PRT452 - Software Engineering Tools**

**TATA RAKESH – S300604**

# Task1

The Git Directory in which the developed code to check the connectivity of a given graph.



The link for the directory is given by,

<https://github.com/tatarakesh/Connected-Graph>

# Task 2

Test cases for the developed program are,

The number of vertices for the given graph =5

The edges of the graph is given by,

(1, 0);

(4, 0);

(1, 2);

(2, 3);

Result

The graph is Connected graph

**Task 3**

# Accidental Bug

In this type of bugs for a program, it may causes by a single stray “:” at the end of the loop structure of a code block in a program.

The example for this kind of bug the example is given by,

# Data clump bug

In this kind of bugs a group of variables declared in a program are passed in a input or output statement with all together or say in a clump throughout different parts of a given program. The group of variables that makes the data clamp are often closely related and is used together from a method, class or a function.

public Person(String fName, String gender, String lName, Integer age){

this.fName = fName;

this.lName = lName;

this.age = age;

this.gender = gender;

}

public void wcmeNewP(){

.printf("Welcome %s %s, a %d-year-old %s f ",firstName, lastName, age, gender);

}

public void working(){

printf("This is %s working hard on %s in %s", firstName);

}

}

# The Middle man code bug

In an object programming paradigm when a class is implemented or exit in order to delegate another class in the package or application it is important real purpose is. Sometimes this is the result of a refactoring of a method in a program and the logic behind the class is moved out in the refactoring process of the class gradually, and at the end only empty shell of a class remains which consequently leads to the extra memory consumption and increased complexity of the task.

One example is, the recursion error. The sample for this error is given below,

void insert(fnode\*& currnt, int valu) {

if (currnt == NULL)

curr = new fnode(valu, NULL);

else if (fnode->valu > valu)

curr = new lnode(valu, currnt->next);

else {

currnt = currnt->next;

insert(currnt, valu);

}

}

# Message chain

The message chain bug exist in a code when a client or method in the program asks for an object for getting the value of another object leading to the chain of objects. In the following sample the function is calling a message chain that has to that it requires to reach the database, extract the company object then get manager-> individual employee's object from there and at the end the salary of the employees . In this case if the structure of storing the employee details changes, the following code will break and fail to retrieve the data.

Empsalary = database.get\_company(com\_name).

get\_mgr(mgr\_name).

get\_tm\_mmber(emp\_name).salary

# Speculative Generality

In OOPS paradigm sometimes programmers implements or handles the special cases or exceptions that are probably never occur when the developed program or application will be executed and used by the users. Thus the methods in the program which are only invoked by their own test methods or data types, will consequently lead to the increased complexity in the code, maintenance of the program and understanding by the other programmers.

# References

FOWLER, M., BECK, K., BRANT, J., OPDYKE, W. and ROBERTS, D. 1999 Refactoring: Improving the Design of Existing Code, Addison Wesley.

SEAMAN, C. B., SHULL, F., REGARDIE, M., ELBERT, D., FELDMANN, R. L., GUO, Y. and GODFREY, S. 2008 Defect categorization: making use of a decade of widely varying historical data. Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement. Kaiserslautern, Germany, ACM.

MENS, T. and TOURWE, T. 2004 A survey of software refactoring. Software Engineering, IEEE Transactions on, 30, 126-139.