International Journal of Computer (IJC)

ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

Code Writing Problem in Python Programming Learning Assistant System

Hsu Wai Hnin[a]*, Khin Khin Zaw[b]

[a,b]*Department of Computer Engineering and Information Technology, Yangon Technological University,*
Yangon , Myanmar
[a]*Email: hsuwaipyplas01@gmail.com*, [b]*Email: thihakhinkhin85@gmail.com*

Abstract

To advance python programming knowledge of students, Python Programming Learning Assistant System (PYPLAS) is elaborated. Currently, PYPLAS included the element fill-in-blank problem and error debugging problem to improve the student's python programming skill. Then, students learn python testing techniques such as unit test (PYUNIT), PYTEST and coverage, etc. In this paper, PYPLAS provides the code writing problem so that students can learn the python code testing and write the python code according to test code. So, code writing problem is generated by using Test Driven Development (TDD) method. Teacher gives the test code to students. Students answer the program code according to test code by using unit test (PYUNIT). Teacher checks the answer code according to coverage in TDD method. For evaluations, 100 codes are provided to accurate the code testing and coverage approach in TDD method. 11 problems are generated and asked 5 learners from training center and universities to solve them in offline PYPLAS. Their results are checked by using coverage approach to display their correct rate. The results show the code writing problem is helpful to test the student's coding quality.

Keywords: PYPLAS; fill-in-blank problem; error debugging problem; unit test; coverage; PYUNIT; TDD; code writing problem.

1. Introduction

Software bugs can cause problems in business, social workplace, and network communications. It is important for software companies to make sure there are no software bugs. So, software testing must be done carefully when writing software. By doing software testing, you need to make sure that the software requirements are met and that the software is secure[1,2]. Nowadays, python programming is one of the most widely used software for business, societies, artificial intelligence, web development and game and research fields. Currently, universities teach python programming to support research fields [3,4].

----

* Corresponding author.

To improve student's programming skill, Python Programming Learning Assistant System (PYPLAS) is developed. Currently, PYPLAS provides, element fill-in-blank problem, error debugging problem [5,6]. In most universities over the world, besides the programming languages, the teachers also teach program testing techniques to improve the testing of the software quality. Then, they teach the unit test and PYTEST that is the basic testing of python testing. Students carefully study python testing techniques to improve their programming testing knowledge [7, 8, 9]. In PYPLAS, a code writing problem is created that aided for the learning of python testing techniques as our contribution. The code writing problem uses unit test, PYTEST and coverage techniques to enhance the coding testing knowledge by using test-driven development (TDD) method. In code writing problem, the teachers provide test code questions to students via USB. The students look at test code questions, write the source code, and check with PYTEST until the source code is correct. The students provide the source code answer files to the teacher. The teacher checks the student's answers code with TDD method that includes testing the correctness and coverage of source code. For correctness, 100 test code questions were extracted from 11 different types of python programming and the 100 answer codes were testing using the coverage technique in TDD method. For evaluations, the five learners from training center and universities answer the 11 problems. The average coverage rate is 82% of all problems and the standard deviation is 0.15 of all problems by answering 5 learners. The main purpose of this research is to generate the code writing problem by using test driving development method (TDD) and to support the student programming skill in Myanmar.

2. Methodology

In this section, the methodology is described such as test driven development, unit test, pytest, coverage method. These methods are used to generate the code writing problem in python programming learning assistant system (PYPLAS).

2.1. Test Driven Development Method

The test driven development method is used to generate the code writing problem in PYPLAS. By using TDD method, the test code is written to check the source code requirements. TDD method repeatedly checks the source code and implements the source code to obtain the reliability source code during its development [10, 11]. The basic cycle in the TDD method is as follows:

- To write the test code to check each function in a source code according to require specifications.
- To write the source code according to the test code
- To repeat modifications of the source code until it success each function of test code. The following Figure 1 shows these steps of TDD method.

2.2. Unit Test Method

In code writing problem, unit test (PYTEST) is used as a python unit testing framework to support the TDD method. Unit test performs test automation, sharing or setup and shutdown code for tests, aggregation of tests into collections, and independence of the tests from the reporting framework. In unit test, a test is firstly written the import the unit test module. Then, a test is created a test case by sub classing ‘unittest.TestCase’. Then the name of the method must start with ‘test’. There are many types of asserts so that a test calls assert functions of TestCase class [8, 12, 13]. This paper uses the ‘assertEqual’ and other asserts method to compare the execution result of the source code.

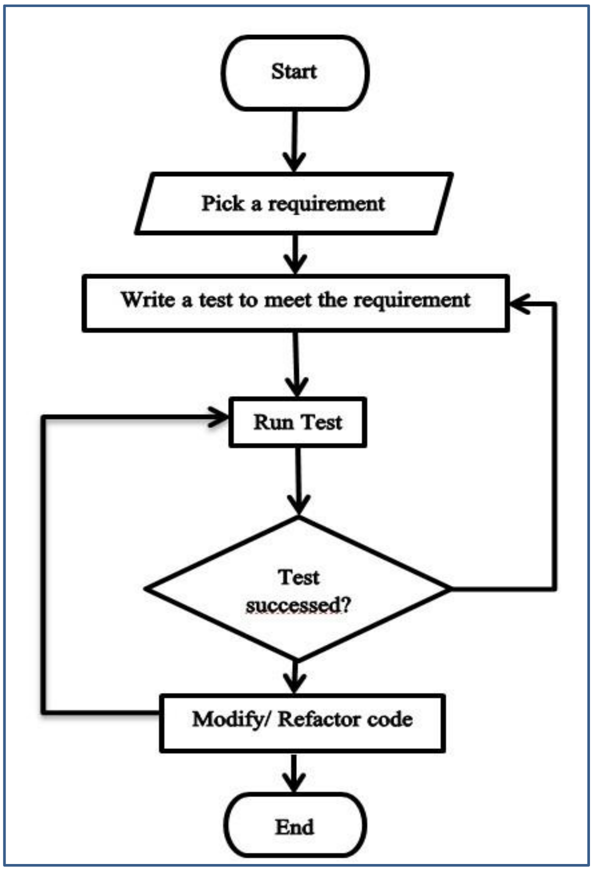
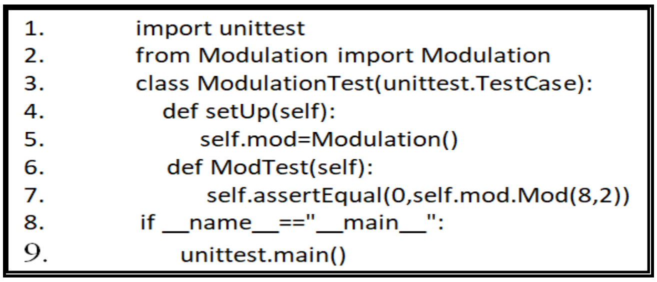


Figure 1: test driven development process.

2.3. Test Code Generation

A test code is written using libraries in PYUNIT (unit test). In test code, we describe how to write an answer source code [14, 15]. Then, the following Figure 2 shows the example of test code 1 for the code writing problem.



```
import unittest
from Modulation import Modulation
class ModulationTest(unittest.TestCase):
   def setUp(self):
       self.mod=Modulation()
   def ModTest(self):
       self.assertEqual(0,self.mod.Mod(8,2))
if __name__=="__main__":
     unittest.main()
```

Figure 2: test code 1.

2.4. Pytest Method

In code writing problem, students could test their answer source code and the teacher could test to check the answer code by using PYTEST to execute their code. PYTEST is a command line utility. PYTEST helps automate run and debug student's code. PYTEST has integration with Python's built-in debug tool, PDB [12, 16, 17]. PYTEST provides many features as part of its ability to execute student's code. PYTEST displays the testing time and error of source code.

2.5. Coverage Method

In code writing problem, teachers use coverage of TDD method to execute the student's source code. The coverage method has three parts of execution source code. Firstly, teachers run the test code by using coverage method to execute the student's code. Then, teachers write the command 'coverage report-m' to display the output such as the statement, miss line of code, and percentage of coverage [9, 18, 19]. Finally, teachers write again the command line 'coverage HTML' to see the HTML page of student's source code results. The coverage method displays the executing time, miss line of code, total statements and percentage of coverage.

3. Code Writing Problem Generation In Pyplas

The purpose of the code writing problem in PYPLAS is developed to help the student for code writing and understand software testing. Python programming uses unit test in TDD method which is the basic software testing technique. So, the unit test is used to teach the student for understanding code testing technique in universities [16, 20, 21]. In offline PYPLAS, teachers wrote the test code questions by using unit test. In teacher view of offline PYPLAS, the teachers can check their test code questions to ask the students. In offline PYPLAS, the code writing problem generation view displays in Figure 3.

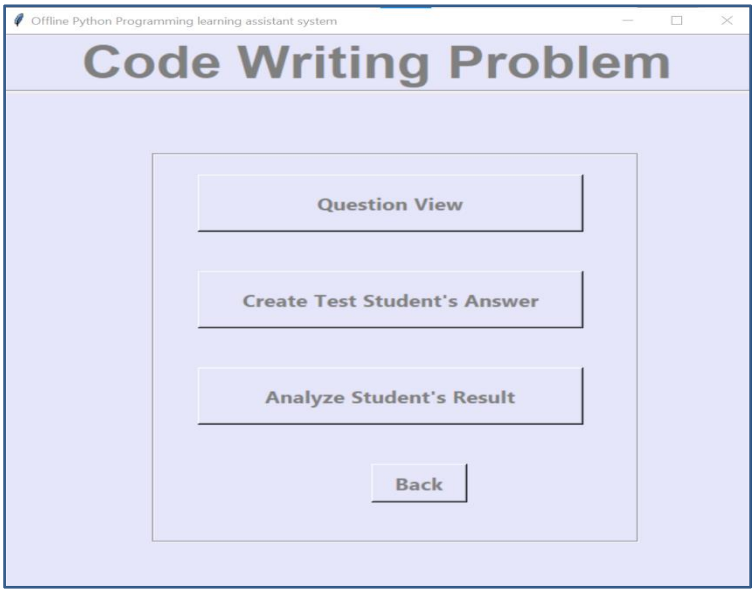
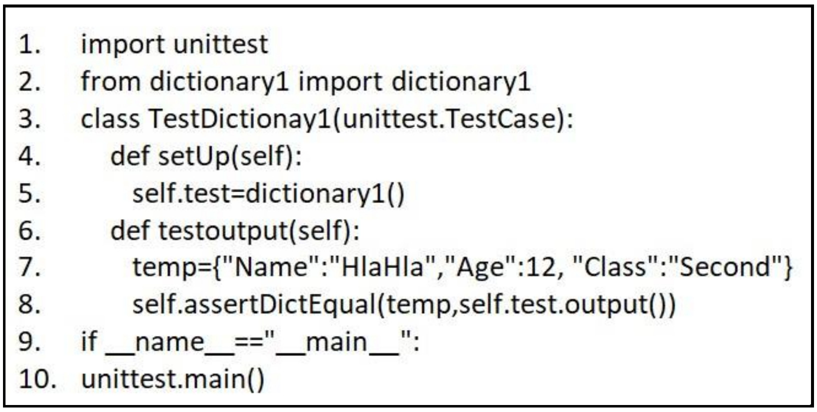

Offline Python Programming learning assistant system
Code Writing Problem
Question View
Create Test Student's Answer
Analyze Student's Result
Back

Figure 3: code writing problem in offline PYPLAS.

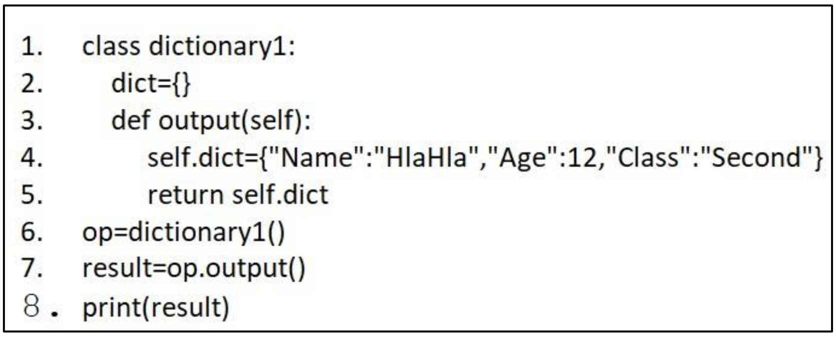
The test code questions are corrected to give the students to test their code testing techniques. Then, the following test code 2 in Figure 4 shows the question of code writing problem.



```
import unittest
from dictionary1 import dictionary1
class TestDictionay1(unittest.TestCase):
   def setUp(self):
     self.test=dictionary1()
   def testoutput(self):
     temp={"Name":"HlaHla","Age":12, "Class":"Second"}
     self.assertDictEqual(temp,self.test.output())
if __name__=="__main__":
unittest.main()
```

Figure 4: test code 2.

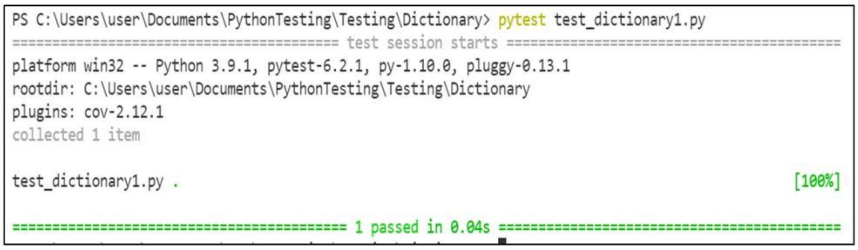
The class name ‘TestDictionary1’ in the test code 2 describes the class name dictionary1 in the source code 2. The method name ‘testoutput’ in the test code describes the ‘output’ method in source code. The name in the test code is related to the name in the source code. In test code 2, the ‘assertDictEqual’ method is used to compare the two dictionaries. Then, the source code 2 is shown in following Figure 5.



```
class dictionary1:
   dict={}
   def output(self):
      self.dict={"Name":"HlaHla","Age":12,"Class":"Second"}
      return self.dict
op=dictionary1()
result=op.output()
print(result)
```

Figure 5: source code 2.

In the source code 2, the class name is ‘dictionary1’ according to test code. Then, the method name is defined ‘output’ according to test code method name ‘testoutput’. This method returns the dictionary and displays the result. For example, the students answer the source code 2 according to the test code 2 questions. Students need to understand the test code written in unit test. Students must rewrite the source code based on the test code questions written in the unit test. When students check the source code using PYTEST, they have to rewrite it repeatedly until it is correct. Students give answer codes to the teacher via USB stick. Firstly, the teachers use the offline PYPLAS to check the student answer by using PYTEST. The following Figure 6 shows the testing result between test code 2 and source code 2 by using PYTEST.



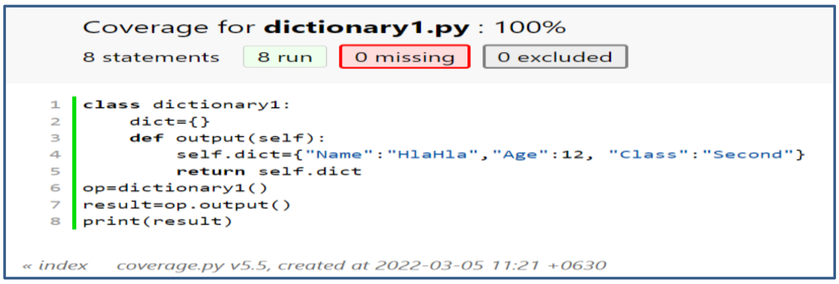
```
PS C:\Users\user\Documents\PythonTesting\Testing\Dictionary> pytest test_dictionary1.py
========================================= test session starts ==========================================
platform win32 -- Python 3.9.1, pytest-6.2.1, py-1.10.0, pluggy-0.13.1
rootdir: C:\Users\user\Documents\PythonTesting\Testing\Dictionary
plugins: cov-2.12.1
collected 1 item

test_dictionary1.py .                                                                            [100%]

========================================== 1 passed in 0.04s ===========================================
```

Figure 6: testing result of test code 2 and source code 2.

Teachers use the coverage technique in TDD method to check the student's answer code. When teachers check the answer code using the coverage method, she will see the wrong line in red and the correct percentage in HTML page. The following Figure 7 shows the results of student's source code.



Coverage for **dictionary1.py** : 100%

8 statements | 8 run | 0 missing | 0 excluded

```
class dictionary1:
    dict={}
    def output(self):
        self.dict={"Name":"HlaHla","Age":12, "Class":"Second"}
        return self.dict
op=dictionary1()
result=op.output()
print(result)
```

« index   coverage.py v5.5, created at 2022-03-05 11:21 +0630

Figure 7: coverage testing results of the student's answer code.

4. Results and Discussion

4.1. Analysis of Code Writing Problem

The analysis of code writing problem is described. For evaluation, 100 test codes are generated and analysis this test code by PYUNIT. Firstly, teacher generated the 100 test codes and checked this code by using unit test. Then, teacher checked these codes correctness by answering these questions before delivering to the students. Table 1 shows the analysis of 100 answer codes according to the 100 test codes in 10 categories. It shows the executing times, line of statements, missing lines, percentage of coverage of each answer code. The average number of coverage rate is 100% so that the total number of missing statements is 0. If the average coverage rate is 44.4%, the number of missing statements is 125. For average number of 100 answers code, average coverage is 72.98% and the missing statements is 4.36%. The standard deviation of coverage in the 10 category is 0.076. It is found that all test codes are working and testing the answer codes of the teacher correctly. The test code is used for testing the student programming skill because of the missing statements is less than 5%. This test code questions is used to test the students who are correctly answered these questions.

Table 1: Analysis of 100 codes.

Category	No	Total Number of statements	Total number of missing statements	Average Coverage	Standard deviation of coverage rate
Operator	10	94	125	44.4%	0.071
String	10	120	41	65.4%	0.04
Decision	10	173	42	75.7%	0.02
Dictionary	10	140	36	76.1%	0.13
List	10	138	48	64.8%	0.05
Tuple	10	86	0	100%	0
Math	10	129	42	67.2%	0.07
Class and obj	10	144	38	73.2%	0.16
File	10	133	21	86.6%	0.25
Try/Catch/Exception	10	162	43	76.4%	0.14
Average number of 100 answer code		13.19%	4.36%	72.98%	
Standard deviation of coverage in the10 category					0.076

4.2. Analysis of Solution Performance

To evaluate performance of students, 11 test codes are chosen from 100 test codes. 5 learners are asked in university and they answered the 11 test codes of code writing problems. Then, answer codes of 5 learners are tested with the PYTEST and coverage in TDD method as evaluation. T1 is for testing the student's code which is correct or not by using PYTEST. T2 is the testing for the coverage of student's code in Table 2. This table shows the average correct rate of each learner and Learner 1 have more correct rate than other learners. Learner 4 and Learner 5 have the correct rate less than other learners. So, they must mostly study the python testing techniques.

Table 2: Average correct rate of each learner.

Number of problem	L1		L2		L3		L4		L5	
	T1	T2	T1	T2	T1	T2	T1	T2	T1	T2
Problem1	pass	100%	pass	56%	pass	78%	pass	57%	pass	56%
Problem2	pass	100%	pass	100%	pass	58%	pass	55%	pass	55%
Problem3	pass	100%	pass	100%	pass	69%	pass	80%	pass	80%
Problem4	pass	100%	pass	100%	pass	100%	pass	67%	pass	67%
Problem5	pass	100%	pass	100%	pass	100%	pass	79%	pass	79%
Problem6	pass	100%	pass	69%	pass	100%	pass	71%	pass	64%
Problem7	pass	100%	pass	70%	pass	100%	pass	70%	pass	70%
Problem8	pass	100%	pass	57%	pass	100%	pass	57%	pass	57%
Problem9	pass	100%	pass	100%	pass	100%	pass	67%	pass	67%
Problem10	pass	100%	pass	100%	pass	100%	pass	72%	pass	72%
Problem11	pass	100%	pass	100%	pass	100%	pass	67%	pass	67%
Average correct rate		100%		87%		91%		67%		67%

The performance for each learner is as shown in Table 3. The teacher checks the answer codes of L2 learner is 9.844s as long time and the answer codes of L1 learner has 0.93 as short time. L1 learner has 101 statements, zero missing statement, 100% average coverage rate and 0.93s run time for 11 answer code because of L1 has more knowledge of python programming and writing coding skill. L5 learner has 122 statements, 39 missing statements, 66.72% average coverage rate and 3.4s run time for 11 answer codes because of L5 has low level of python programming knowledge and writing skill. So, L5 learner has more learning the python programming. In Table 3, the average coverage rate of 5 learners is 82%. The standard deviation of 5 learners with coverage rate is 0.15.

Table 3: Solution performance of learner

Learner id	Statements of 11 answer code	Missing statements of 11 answer code	Average coverage rate	Run time for answering each learner
L1	101	0	100%	0.39s
L2	117	14	87%	9.844s
L3	108	13	91%	0.39s
L4	123	35	67%	0.43s
L5	122	39	67%	3.4s
Average			82%	
Standard deviation			0.15	

The average coverage rate of each learner is shown in Figure 8. This figure shows the Learner 1 is more correct rate and Learner 5 is the less correct rate than other learners.

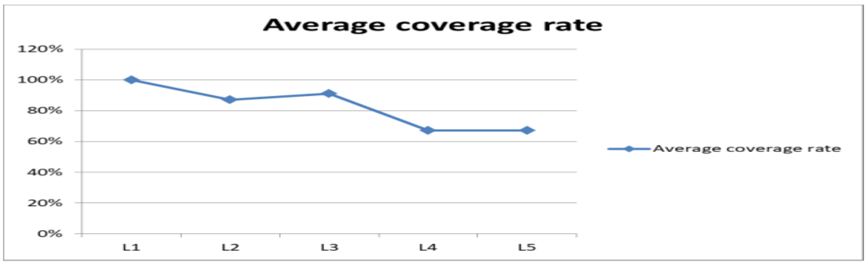


Figure 8: The average correct rate of learners.

After the answering, we asked the 5 learners to reply their suggestion and thinking for code writing problem in python programming education according to the five questions in Table 4.

Table 4: Questions for questionnaire.

No	Question
Q1	Do you think this system is easy?
Q2	Do you think this system is supported to check the program functions?
Q3	Do you think the test code is easy to understand as the assignment?
Q4	Do you think the answering time is long?
Q5	Do you think this system is helpful in understanding python testing?

Learners replied to the five questions in Table 5. For Q1, L1, L2, L3 learners suggested that this code writing problem is moderate for students because of the test code is not easy to understand. L4 and L5 learners are hard to understand these questions for answering the students. So, the easy questions of code must be written to ask the students. Then, their suggestions give this problem which is useful for student to learn python testing techniques. In future, modification of this problem is to ask the students.

Table 5: Questionnaire results with each learner.

ID	Q1	Q2	Q3	Q4	Q5
L1	moderate	useful	easy	short	helpful
L2	moderate	useful	easy	long	helpful
L3	moderate	useful	easy	long	helpful
L4	hard	useful	easy	long	helpful
L5	hard	useful	easy	long	helpful

5. Conclusion

In this paper, we proposed a code writing problem in Python Programming Learning Assistant System (PYPLAS). We produced the code writing problem by using the test-driven development (TDD) method. It is intended to support the python programming testing courses in educational effects. In code writing problem, the source code and the test code can be tested automatically by using the open source software unit test. Then, the coverage is used to test the student's coverage correct rate and missing statement. For evaluation, 100 test codes are analyzed by using PYUNIT and Coverage technique. Then, 11 test codes assign the five learners to analysis the correct rate of student's answer code and calculate the coverage rate and missing statements of student's answer code. In the future, we must ask the students and analysis the student's coverage rate. Then, we must upload the code writing problems in online PYPLAS web server to automatically test the answer codes and to reduce the workload of teacher by using PYTEST and coverage.

Acknowledgement

I would like to be appreciative of Dr.Khin Khin Zaw by learning their research and supporting me to perform this research under the supervision and discussion. I would like to thank all people who participate to answer questions of code writing problem in this paper.

References

[1] K.R.Srinath. "Python-The Fastest Growing Programming Language". International Research Journal of Engineering and Technology (IRJET), vol. 04, pp.354-357, Dec.2017.

[2] S.R.MD.Derus, A.Z.M.Ali. "Difficulties in learning programming: Views of students". in Proc. ICCIE, 2012, pp.1-6.

[3] K.Adawadkar. "Python Programming-Applications and Future". International journal of Advance Engineering and Research Development.vol.4, pp.1-4, Apr.2017.

[4] P.K.Karmore, G.L.Girhe. "Programming language python: A review". IJARIIE [on-line].6(2), pp.1634-1637. Available: http://ijariie.com [2020].

[5] B.R.Payne. “Learning to python in python-by teaching it”. in Proc.ISTEMTLC, 2017, pp.99-106.

[6] H.W.Hnin and K.K.Zaw. “Element Fill-in-blank problems in Python Programming Learning Assistant System”. in Proc.ICAIT, 2020. pp.88-93.

[7] S.S.Ghuman. “Software testing techniques”, International Journal of Computer Science and Mobile Computing, vol.3, pp.988-993, Oct.2014.

[8] S. lukasczyk, F. Kroik, and G. Fraser. “Automated Unit Test Generation for Python”, IEEE. [on-line], Available: www.spectrum .ieee .org/ computing/ software. [Jul.25, 2020].

[9] M.Thakur, Sanjay. ‘Review on Structural Software Testing Coverage Approaches”, International Journal of Advance research Ideas and Innovations in Technology, vol.3, pp.988-993, Oct.2014.

[10] N.Funabiki, Y.Matsushima, T.Nakanishi, K.Watanabe, and N.Amano. “A Java Programming Learning Assistant System Using Test-Driven Development Method”, IAENG. [on-line].40(1). Available: www.iaeng.com[Feb.9,2013].

[11] D.Sale. “Driving Your Development with Test”, in Testing Python Applying Unit Testing, TDD, BDD, and Acceptance Testing, 1th ed, vol.1.R. Mengle, Ed. United Kingdom: John Wiley & Sons, 2014, pp.68-83.

[12] M.E.Khan, Khan. “Important of Software Testing in Software Development Life Cycle”, International journal of Computer Science, vol.11, pp.120-123, Mar.2014.

[13] A.Orso, G.Rothermel. “Software Testing: A Research Travelogue”, in Proc.FOSE’14, 2014, pp.1-16.

[14] S.M.Salleh, Z.Shukur, H.M.Judi. “Analysis of Research in Programming Teaching Tools: An Initial Review”, in Proc.Social and Behavioral Sciences, 2013, pp.127-135.

[15] D.Young. (2015, March 9) Software Testing Overview. (1th edition). [on-line].12(1). Available: www.researchgate.net/publication/273319104[Mar 9, 2015].

[16] X.kui, W.Liu, H.Du. “Research on the Improvement of Python Language Programming Course Teaching Methods based on Visualization”, in Proc. ICCSE, 2017, pp.639-644.

[17] X.Shi, Y.Chen. “New Teaching Method of Python Programming for Liberal Arts Students”, IJIRES [on-line].7(3), pp.261-271. Available: www. IJIRSE.com [Jun.30,2020].

[18] A.Sharma, D.Sharma, S.Gupta. “Python : The Programming Language of Future”, IJIRT. [on-line].6(12),pp.115-118. Available: www. IJIRT.com[May.3,2020].

[19] U.Nikula, J.Sajaniemi, M.Tedre. “Python and Roles of Variables in Introductory Programming: Experiences from Three Educational Institutions”, Journal of Information Technology Education,vol.6,pp.200-214,May.2007.

[20] T.L.Chen, T.C.Hsiao, T.C.Kang, T.Y.Wu, C.C.Chen. “Learning Programming Language in Higher Education for Sustainable Development: Point-Earning Bidding Method”, Sustainability. [on-line].12(4), pp.2-14. Available: www.sustainability.com [Jun.1,2020].

[21] A.L.S.Saabith, M.M.M.Fareez.T.Vinothraj. “Python Current Trend Application-an Overview”, International Journal of Advance Engineering and Research Development, vol.6,pp.6-11,Oct.2019.