Bank Management System JUnit Testing

Submitted To

Mr. Abdus Satter

Lecturer

Institute of Information Technology,

University of Dhaka

Submitted By

Muhabbat Sarker Eshan

BSSE 0939

Date of Submission

September 07, 2019

Institute of Information Technology

University of Dhaka

LETTER OF TRANSMITTAL

8th September, 2019
To,
Mr. Abdus Satter,
Lecturer,
Institute of Information Technology
University of Dhaka.
Subject: Submission of JUnit testing report on Bank Management System.
Sir,
With due respect, I am submitting the report on the above topic. In this report, I have given my best effort albeit some shortcomings.I earnestly hope that you would excuse all errors and oblige thereby.

Yours sincerely
Muhabbat Sarker Eshan–BSSE 0939

3nd Year, 6th Semester, 9th Batch
Institute of Information Technology
University of Dhaka
Session: 2015-16

Abstract

This report initially describes the short project description of bank management system.Then describes test planning , test design, test technique and finally test case generation and test specification of each test case.

Contents

Table lists

Chapter 1: Test Planning

Project description

The selected project is Bank Management System.Almost 30 to 35 methods are present in this project. These methods carries double and String type parameter and also their return types are String, double and boolean.

Tasks of the project are-

1. To create a bank account
2. Select Account types (Current Account and Savings Account)
3. To create unique PIN and Account No for each user
4. Store information to a text File
5. Transfer of money
6. Deposit money
7. Withdraw money
8. Pay Bill
9. Change PIN

Number of Test cases and Their Cost:

I have created more than 120 test cases for this project.
As the project is very simple so i didn’t cost much to test this project.

Area of Risks:

1. Defect rate: The expected defect rate has been increased due to lack of available requirement specification document.
2. Number of Test Cases: Due to lack of skills and experience I probably could not generate the sufficient number of test cases.

Test completion:

If the project has less than 20% failure then the project is successful. Due to lack of experience Itested the project with unit testing. There’s more ways for testing upon which the completion depends.

Chapter 2: Test Design

The main objective of testing the project is to check the operation inside the methods are fully correct or not. Testing any exception, error, failure occurrence of a methods. And also test is there any boundary value problem occurrence of a method.

Design Techniques:

The technique for testing this project is Black box testing. I have used-

- Equivalence class testing
- Robust testing for the project

Test Objectives:

The main objective of testing the project is to check either users can run various operations on
the system or not and if the operations.Objectives of this testing is to test:

1. Whether user can pay bill from account
2. Whether user can transfer money to another account
3. Whether user can generate unique PIN and account No
4. Whether user can deposit money
5. Whether user can change PIN
6. Whether user can store information

Items to be tested:

1. Creating bank account
2. Adding new account to system
3. Deposit money from both Account(Savings and Current)
4. Withdraw money from both Account(Savings and Current)
5. Transfer money from both Account(Savings and Current)
6. Pay bill from both Account(Savings and Current)
7. Changing PIN of both Accounts
8.Check balance
9. Check minimum balance of an account
10. Check maximum transfer of an account
11. Check withdraw limit

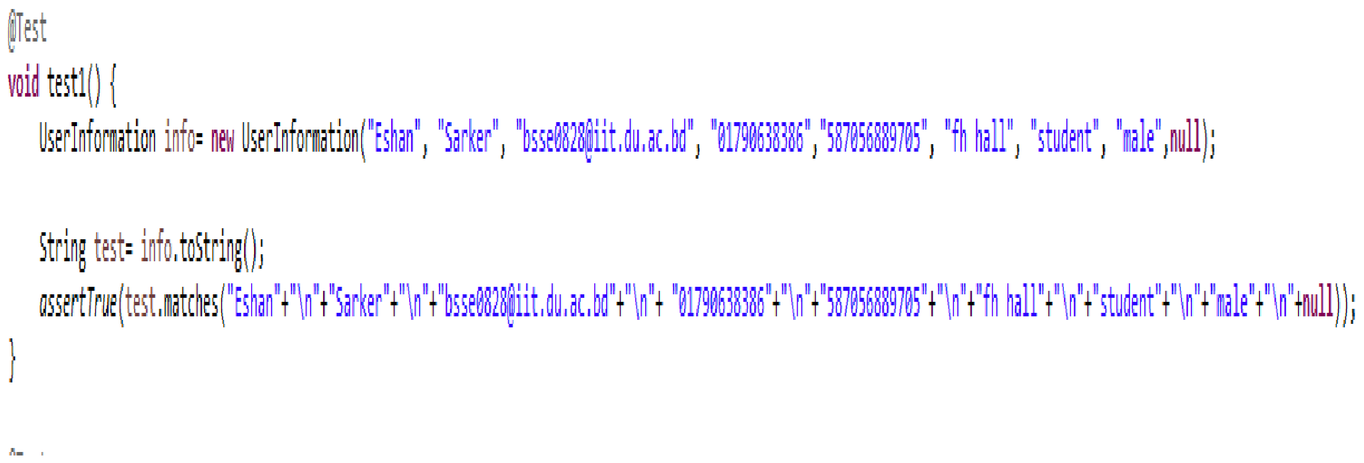
Tools used to generate test cases:

To do unit testing JUnit has been used as a tool.

Chapter 3: Test Execution

Method from UserInformation Class

Test case: T01



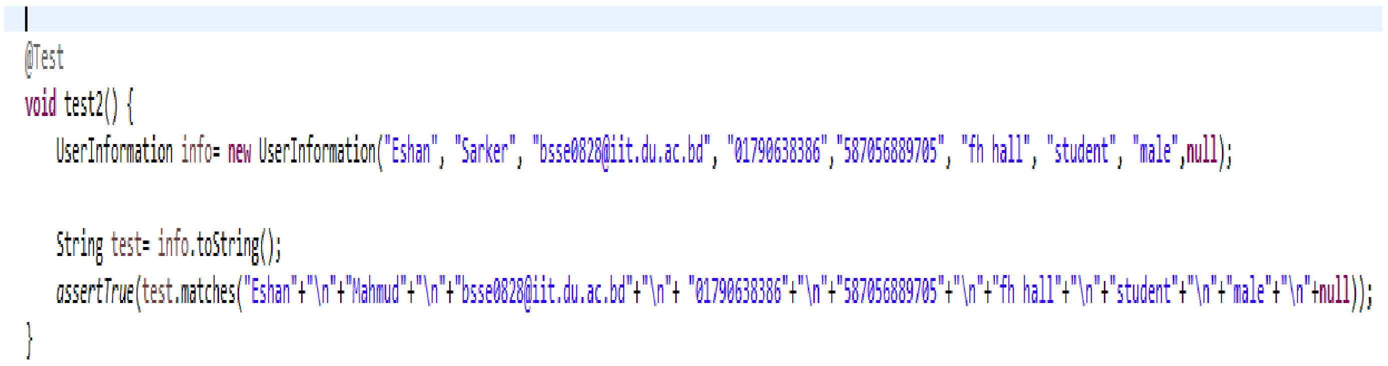
```
@Test
void test1() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    String test= info.toString();
    assertTrue(test.matches("Eshan"+"\n"+"Sarker"+"\n"+"bsse0828@iit.du.ac.bd"+"\n"+ "01790638386"+"\n"+"587056889705"+"\n"+"fh hall"+"\n"+"student"+"\n"+"male"+"\n"+null));
}
```

Figure 1: test case of T01

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T01	Verify that system can take the input for those field	Every input muse be in string	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	passed

Table 1: Test Specification of T01

Test case: T02



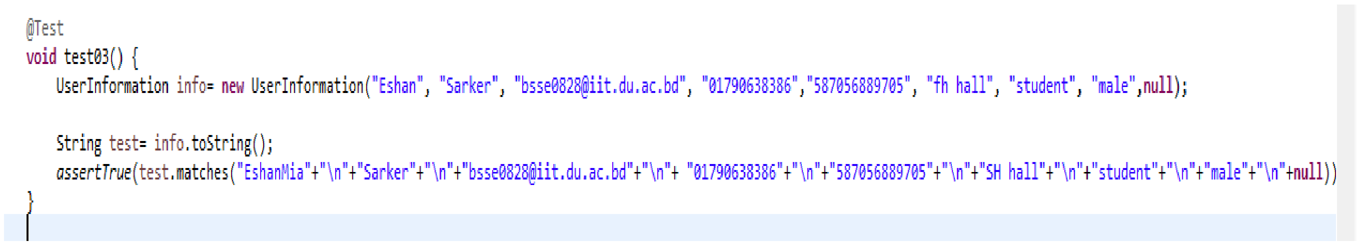
```
@Test
void test2() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    String test= info.toString();
    assertTrue(test.matches("Eshan"+"\n"+"Mahmud"+"\n"+"bsse0828@iit.du.ac.bd"+"\n"+ "01790638386"+"\n"+"587056889705"+"\n"+"fh hall"+"\n"+"student"+"\n"+"male"+"\n"+null));
}
```

Figure 2: test case of T02

Test id	purpose	precond -iton	input	Expected output	Actual output	Test result
T02	Verify that if last name is match or not	Every input muse be in string	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	"Eshan", "Mahmud", "bsse0828@iit.du.ac.bd" , "01790638386","587056889705", "fh hall", "student", "male",null	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	failed

Table 2: Test Specification of T02

Testcase :T03



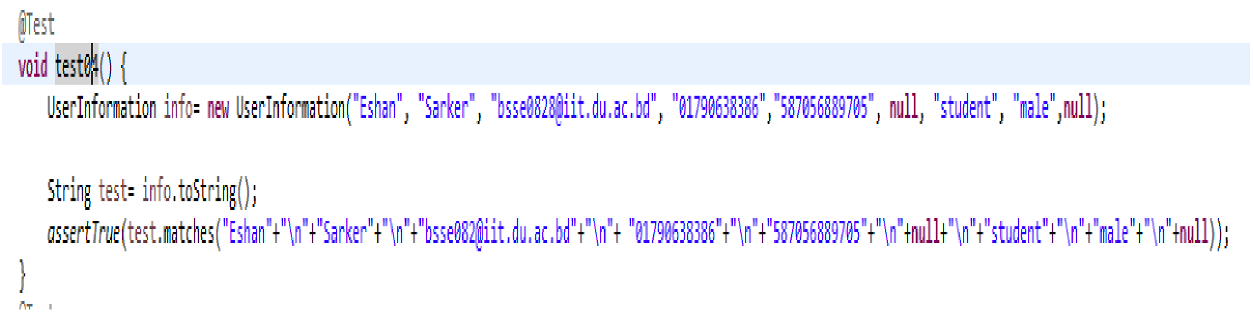
```java
@Test
void test03() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    String test= info.toString();
    assertTrue(test.matches("EshanMia"+"\n"+"Sarker"+"\n"+"bsse0828@iit.du.ac.bd"+"\n"+ "01790638386"+"\n"+"587056889705"+"\n"+"SH hall"+"\n"+"student"+"\n"+"male"+"\n"+null))
}
```

Figure 3: test case of T03

Test id	purpose	precond-iton	input	Expected output	Actual output	Test result
T03	Verify that if address is match or not	Every input muse be in string	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	"Eshan", "Mahmud", "bsse0828@iit.du.ac.bd" , "01790638386","587056889705", "SH hall", "student", "male",null	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	failed

Table 3: Test Specification of T03

Testcase: T04



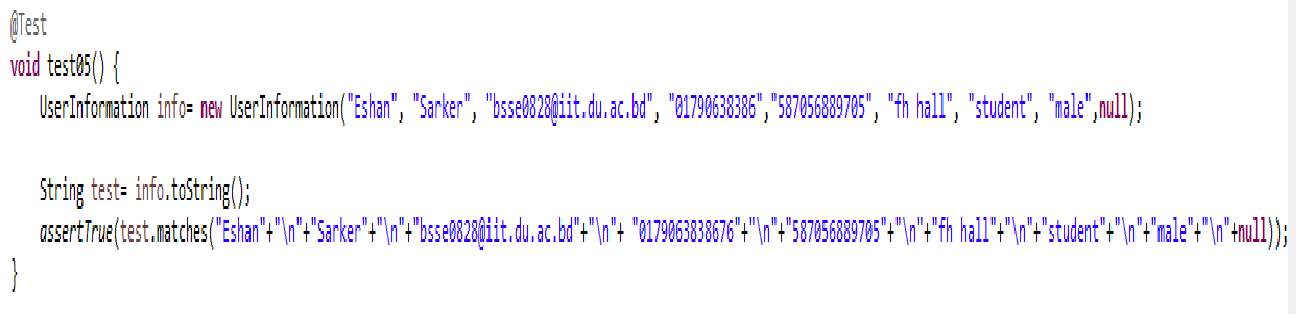
```
@Test
void test04() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", null, "student", "male",null);

    String test= info.toString();
    assertTrue(test.matches("Eshan"+"\n"+"Sarker"+"\n"+"bsse082@iit.du.ac.bd"+"\n"+ "01790638386"+"\n"+"587056889705"+"\n"+null+"\n"+"student"+"\n"+"male"+"\n"+null));
}
```

Figure 4: test case of T04

Test id	purpose	precond -iton	input	Expected output	Actual output	Test result
T04	Verify that if email is match or not	Every input muse be in string	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	"Eshan", "Mahmud", "bsse082@iit.du.ac.bd", "01790638386","587056889705", "SH hall", "student", "male",null	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	failed

Table 4: Test Specification of T04

Testcase: T05



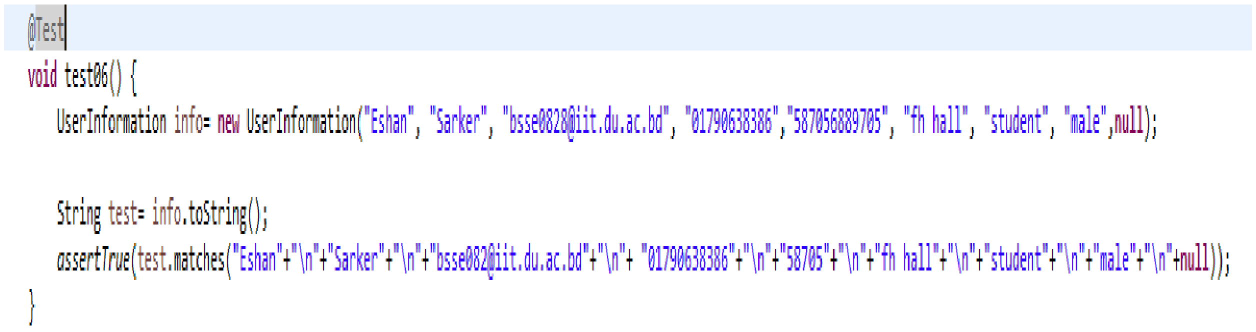
```
@Test
void test05() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    String test= info.toString();
    assertTrue(test.matches("Eshan"+"\n"+"Sarker"+"\n"+"bsse0828@iit.du.ac.bd"+"\n"+ "0179063838676"+"\n"+"587056889705"+"\n"+"fh hall"+"\n"+"student"+"\n"+"male"+"\n"+null));
}
```

Figure 5: test case of T05

Test id	purpose	precond -iton	input	Expected output	Actual output	Test result
T05	Verify that if phone number is match or not	Every input muse be in string	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	"Eshan", "Mahmud", "bsse0828@iit.du.ac.bd" , "01790638376","587056889705", "SH hall", "student", "male",null	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	failed

Table 5: Test Specification of T05

TestCase: T06



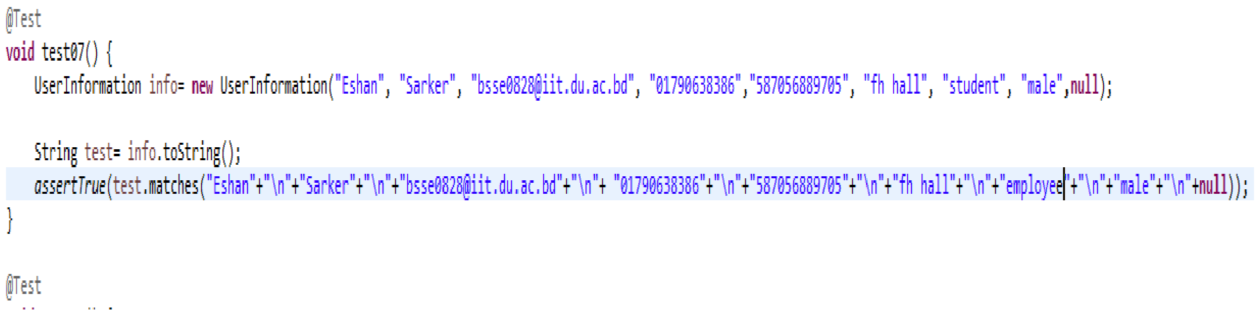
```
@Test
void test06() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    String test= info.toString();
    assertTrue(test.matches("Eshan"+"\n"+"Sarker"+"\n"+"bsse082@iit.du.ac.bd"+"\n"+ "01790638386"+"\n"+"58705"+"\n"+"fh hall"+"\n"+"student"+"\n"+"male"+"\n"+null));
}
```

Figure 6: test case of T06

Test id	purpose	precond -iton	input	Expected output	Actual output	Test result
T06	Verify that if NID number is match or not	Every input muse be in string	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	"Eshan", "Mahmud", "bsse0828@iit.du.ac.bd" , "01790638376","587056889705", "SH hall", "student", "male",null	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	failed

Table 6: Test Specification of T06

Testcase: T07



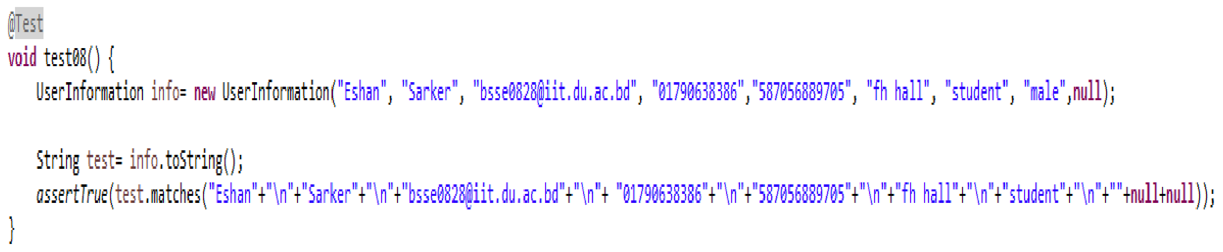
```
@Test
void test07() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    String test= info.toString();
    assertTrue(test.matches("Eshan"+"\n"+"Sarker"+"\n"+"bsse0828@iit.du.ac.bd"+"\n"+ "01790638386"+"\n"+"587056889705"+"\n"+"fh hall"+"\n"+"employee"+"\n"+"male"+"\n"+null));
}

@Test
```

Figure 7: test case of T07

Test id	purpose	precond -iton	input	Expected output	Actual output	Test result
T07	Verify that if occupation is match or not	Every input muse be in string	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	"Eshan", "Mahmud", "bsse0828@iit.du.ac.bd" , "01790638376","587056889705", "SH hall", "student", "male",null	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	failed

Table 7: Test Specification of T07

TestCase: T08



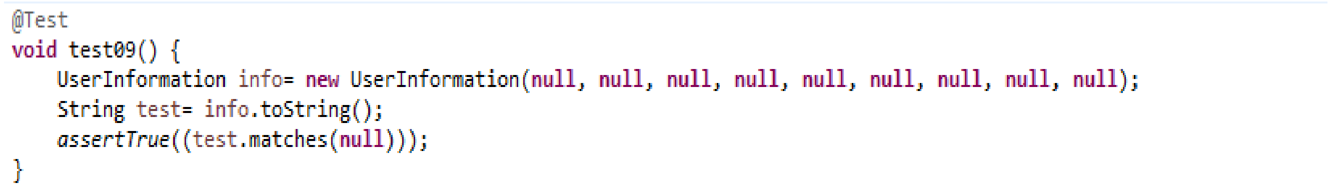
```
@Test
void test08() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    String test= info.toString();
    assertTrue(test.matches("Eshan"+"\n"+"Sarker"+"\n"+"bsse0828@iit.du.ac.bd"+"\n"+ "01790638386"+"\n"+"587056889705"+"\n"+"fh hall"+"\n"+"student"+"\n"+""+null+null));
}
```

Figure 8: test case of T08

Test id	purpose	precond -iton	input	Expected output	Actual output	Test result
T08	Verify that if gender is match or not	Every input muse be in string	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	"Eshan", "Mahmud", "bsse0828@iit.du.ac.bd" , "01790638376","587056889705", "SH hall", "student", "male",null	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	failed

Table 8: Test Specification of T08

TestCase: T09



```
@Test
void test09() {
    UserInformation info= new UserInformation(null, null, null, null, null, null, null, null, null);
    String test= info.toString();
    assertTrue((test.matches(null)));
}
```

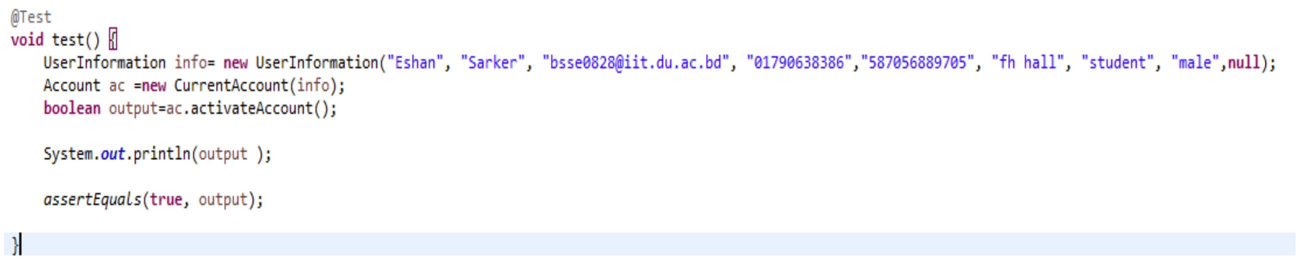
Figure 9: test case of T09

Test id	purpose	precond -iton	input	Expected output	Actual output	Test result
T09	Verify that if system works or not for all null input	Every input muse be in string	null, null, null, null, null, null, null, null, null	"Eshan", "Mahmud", "bsse0828 @iit.du.ac.bd", "01790638376","587056889705", "SH hall", "student", "male",null	"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	error

Table 9: Test Specification of T09

ActiveAccount Method from CurrentAccount

TestCase: T10



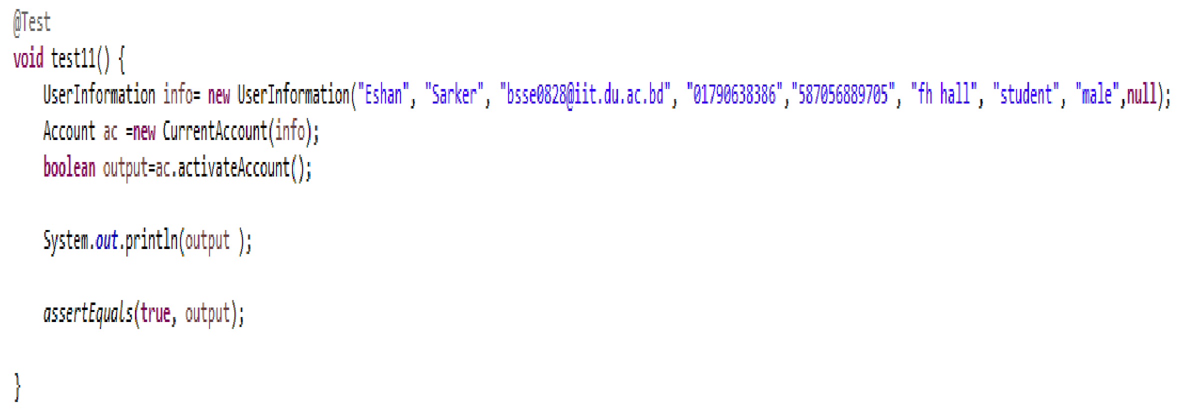
```
@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.activateAccount();

    System.out.println(output );

    assertEquals(true, output);

}
```

Figure 10: test case of T10

Test id	purpose	precondit-on	input	Expected output	Actual output	Test result
T10	To check if the account is active or not		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	true	true	passed

Table 10: Test Specification of T10

Testcase: T11



```
@Test
void test11() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.activateAccount();

    System.out.println(output );

    assertEquals(true, output);

}
```

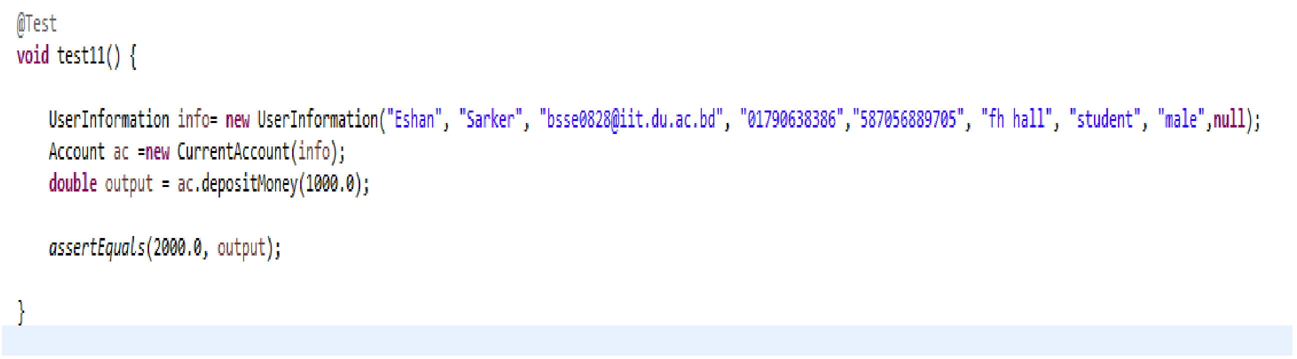
Figure 11: test case of T11

Test id	purpose	precondit-on	input	Expected output	Actual output	Test result
T11	To check if the account is active or not		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	false	true	failure

Table 11: Test Specification of T11

depositeMoney method from CurrentAccount

TestCase: T12



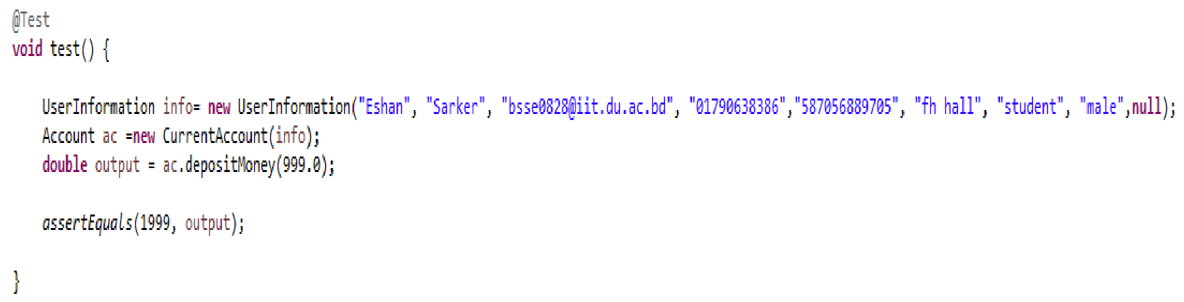
```
@Test
void test11() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output = ac.depositMoney(1000.0);

    assertEquals(2000.0, output);

}
```

Figure 12: test case of T12

Test id	purpose	precondit -on	input	Expected output	Actual output	Test result
T12	To check deposit money	Enter a double value	1000.0	2000.0	2000.0	passed

Table 12: Test Specification of T12

Testcase: T13



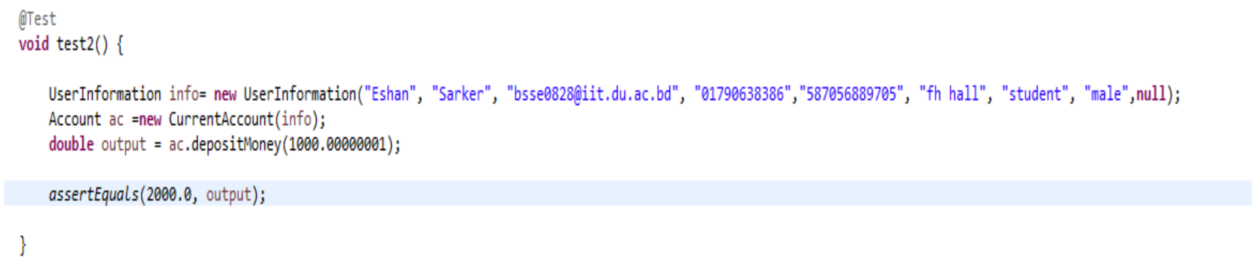
```
@Test
void test() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output = ac.depositMoney(999.0);

    assertEquals(1999, output);

}
```

Figure 13: test case of T13

Test id	purpose	precondit -on	input	Expected output	Actual output	Test result
T13	To check deposit money	Enter a double value	999.0	1999	1999.0	passed

Table 13: Test Specification of T13

TestCase: 14



```
@Test
void test2() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output = ac.depositMoney(1000.00000001);

    assertEquals(2000.0, output);

}
```

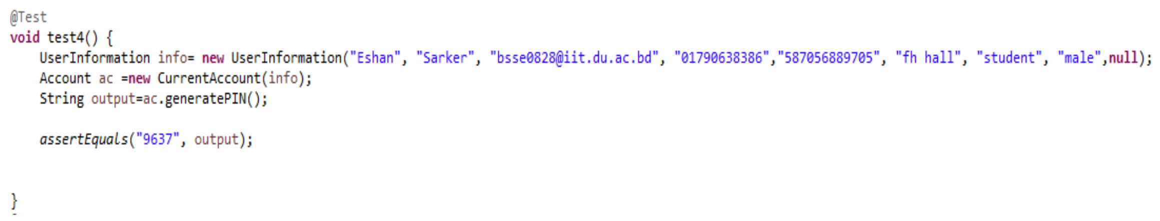
Figure 14: test case of T14

Test id	purpose	precondit-on	input	Expected output	Actual output	Test result
T14	To check deposit money	Enter a double value	1000.00000001	2000.0	2000.00000001	failed

Table 14: Test Specification of T14

generatePIN method from currentAccount Class

TestCase: T15



```
@Test
void test4() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output=ac.generatePIN();

    assertEquals("9637", output);


}
```

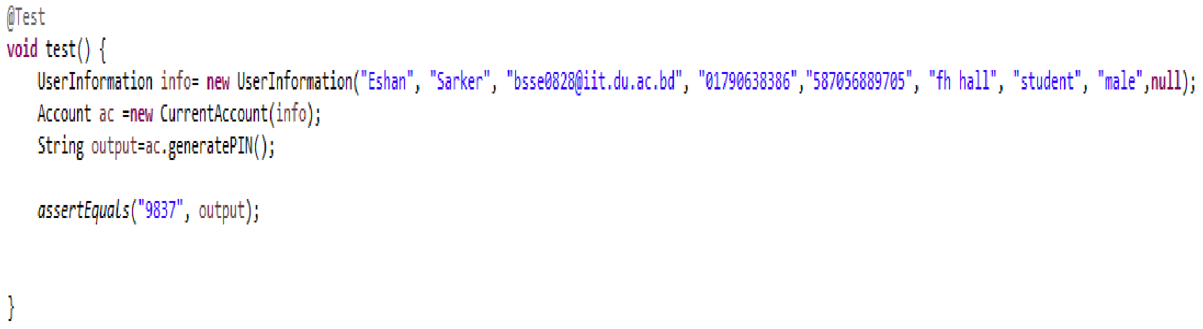
Figure 15: test case of T15

Test id	purpose	precond it-on	input	Expected output	Actual output	Test result

T15	To check if system generate pin every time Or not		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	9637	Generate random number of 4 digits	failed

Table 15: Test Specification of T15

TestCase: 16



```
@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output=ac.generatePIN();

    assertEquals("9837", output);


}
```

Figure 12: test case of T16

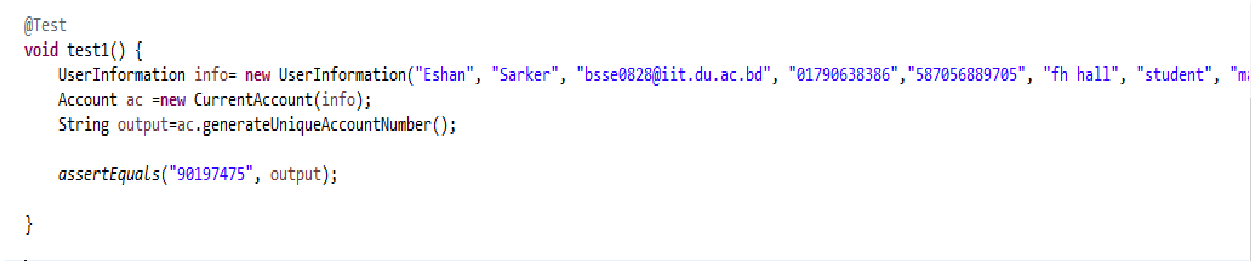
Test id	purpose	precondit-on	input	Expected output	Actual output	Test result

T16	To check if system generate pin every time Or not		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	9837	Generate random number of 4 digits	failed

Table 16: Test Specification of T16

generateUniqueAccountNo method from CurrentAccount

TestCase: T17



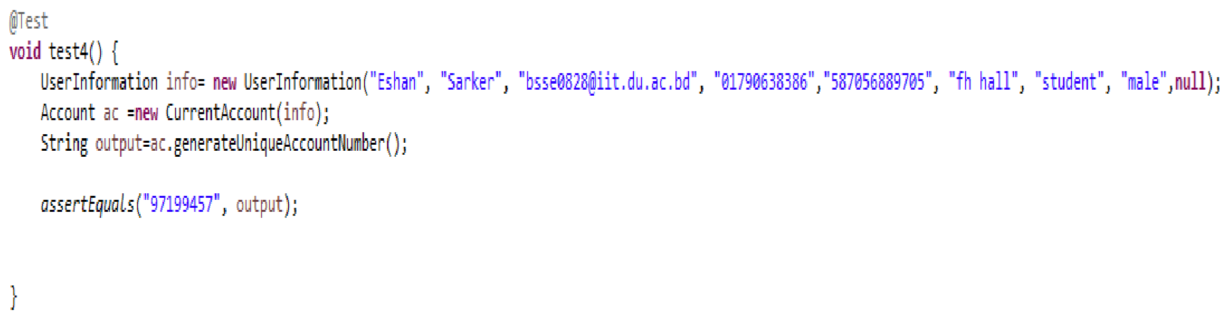
```
@Test
void test1() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "m
    Account ac =new CurrentAccount(info);
    String output=ac.generateUniqueAccountNumber();

    assertEquals("90197475", output);

}
```

Figure 17: test case of T17

Test id	purpose	precondit-on	input	Expected output	Actual output	Test result
T17	To generate unique account no		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","5870568897 05", "fh hall", "student", "male",null	90197475	Generate random number of 4 digits	failed

Table 17: Test Specification of T17

Testcase: T18



```
@Test
void test4() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output=ac.generateUniqueAccountNumber();

    assertEquals("97199457", output);


}
```

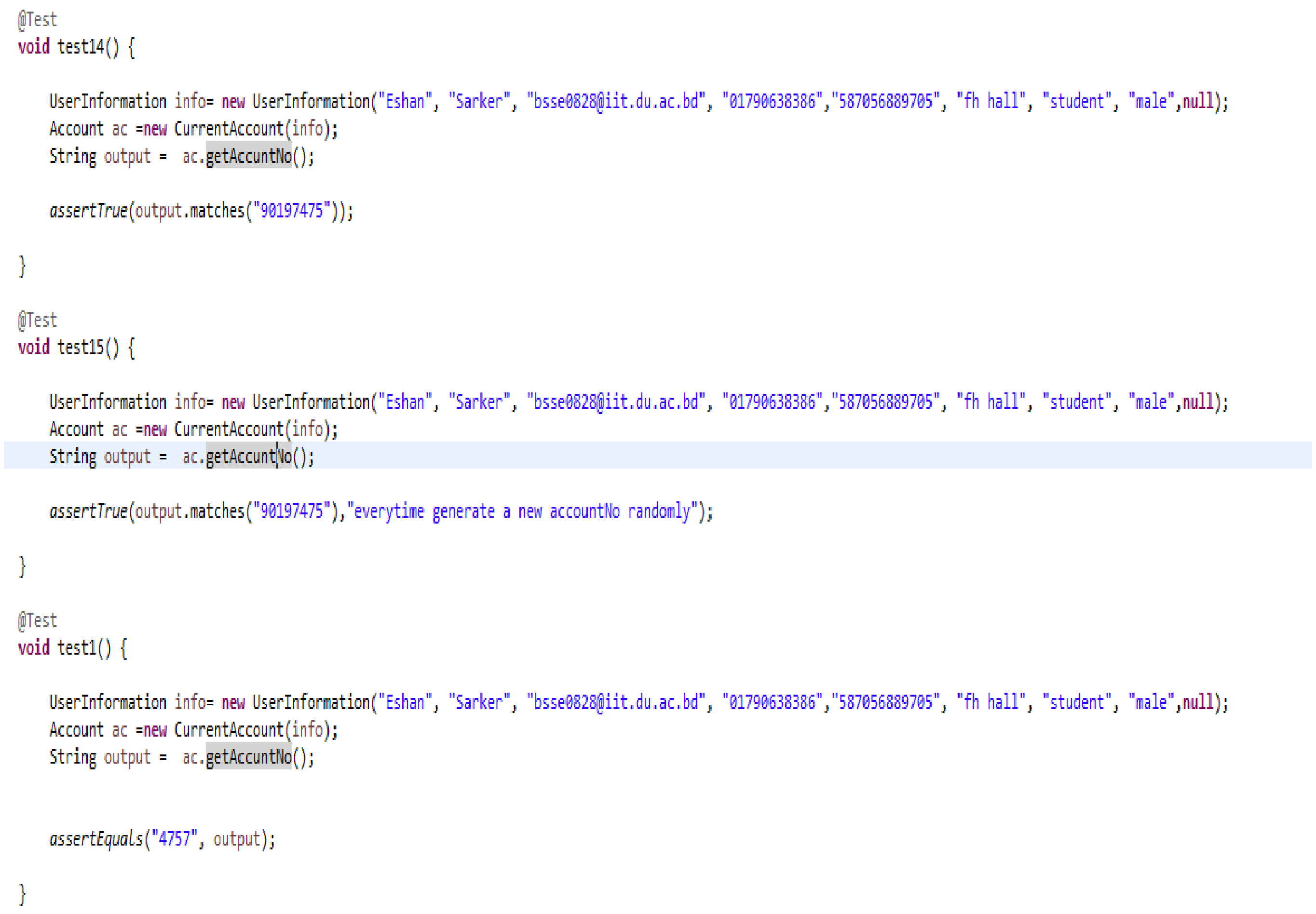
Figure 18: test case of T18

Test id	purpose	precondit-on	input	Expected output	Actual output	Test result
T18	To generate unique account no		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","5870568897 05", "fh hall", "student", "male",null	97199457	Generate random number of 4 digits	failed

Table 18: Test Specification of T18

getAccountNo() method for CurrentAccount Class

TestCase: 19,20,21



```
@Test
void test14() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.getAccuntNo();

    assertTrue(output.matches("90197475"));

}

@Test
void test15() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.getAccuntNo();

    assertTrue(output.matches("90197475"),"everytime generate a new accountNo randomly");

}

@Test
void test1() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.getAccuntNo();


    assertEquals("4757", output);

}
```

Figure 19: test case of T18,20,21

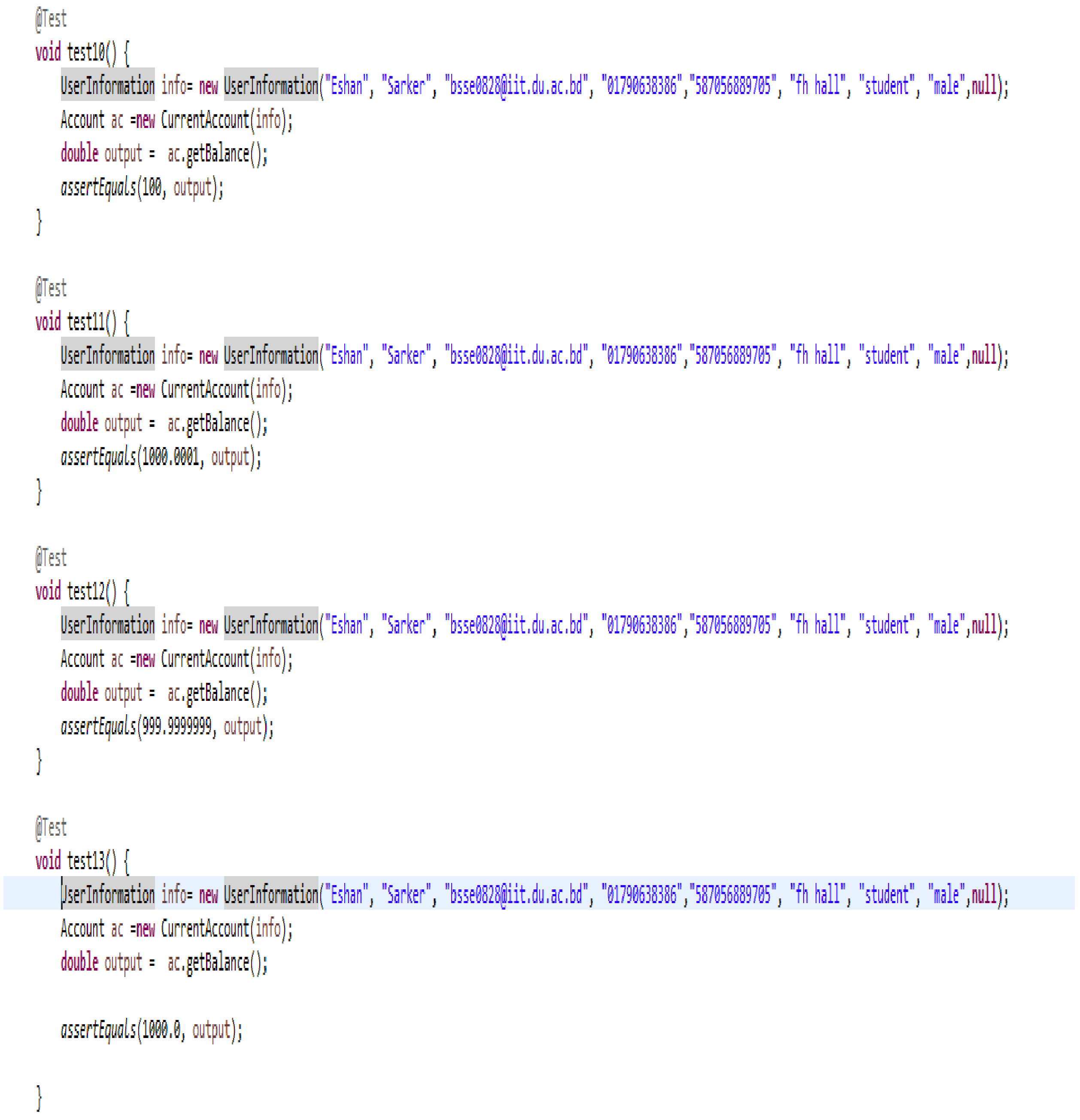
Test id	purpose	precondit-on	input	Expected output	Actual output	Test result
T19	To get account no		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh	90197475	Generate random number	failed

			hall", "student", "male",null			
T20	To get account no		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	4747	Generate random number	failed
T21	To get account no		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	9077475	Generate random number	failed

Table 19: Test Specification of getAccount()

getBallance() Method from CurrentAccount Class

Testcase: 22,23,24,25



```java
@Test
void test10() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output =  ac.getBalance();
    assertEquals(100, output);
}

@Test
void test11() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output =  ac.getBalance();
    assertEquals(1000.0001, output);
}

@Test
void test12() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output =  ac.getBalance();
    assertEquals(999.9999999, output);
}

@Test
void test13() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output =  ac.getBalance();

    assertEquals(1000.0, output);

}
```

Figure 20: test case of T22,23,24,25

Test id	purpose	precondition	input	Expected output	Actual output	Test result

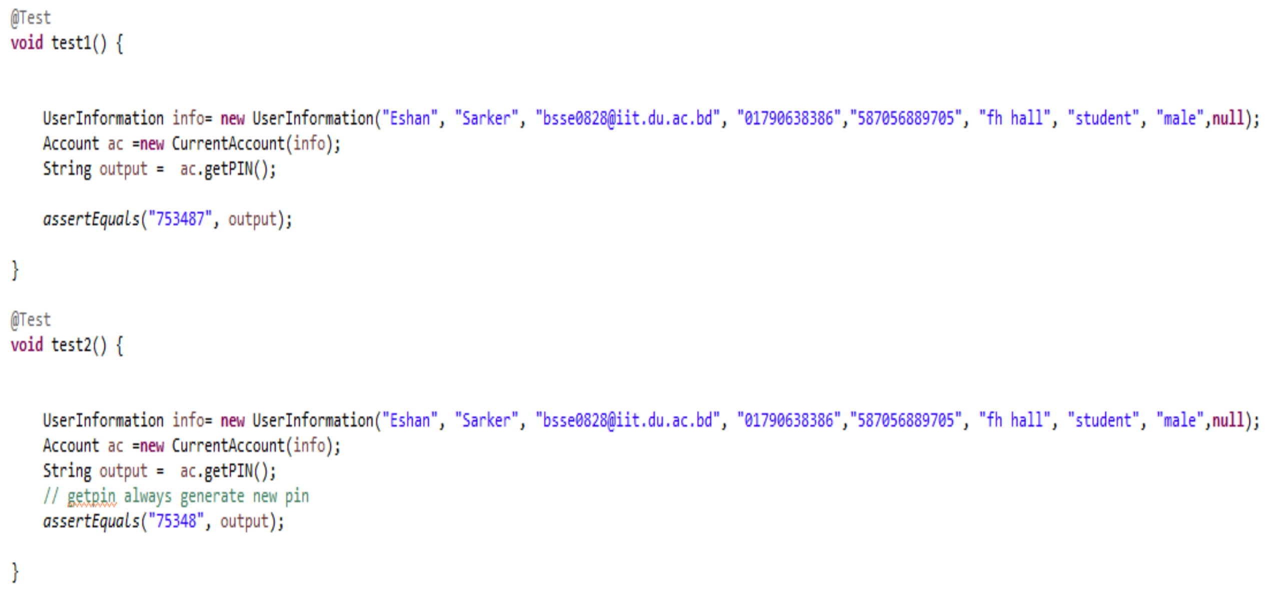
T22	To get balance		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	100	1000.0	failed
T23	To get balance		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	1000.001	1000.0	failed
T24	To get balance		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	999.9999	1000.0	failed
T25	To get balance		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh	1000.0	1000.0	passed

			hall", "student", "male",null			

Table 20: Test Specification of getBallance()

getPIN method from CurrentAccount Class

TestCase: 26,27



```
@Test
void test1() {


    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.getPIN();

    assertEquals("753487", output);

}

@Test
void test2() {


    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.getPIN();
    // getpin always generate new pin
    assertEquals("75348", output);

}
```

Figure 21: test case of T26,27

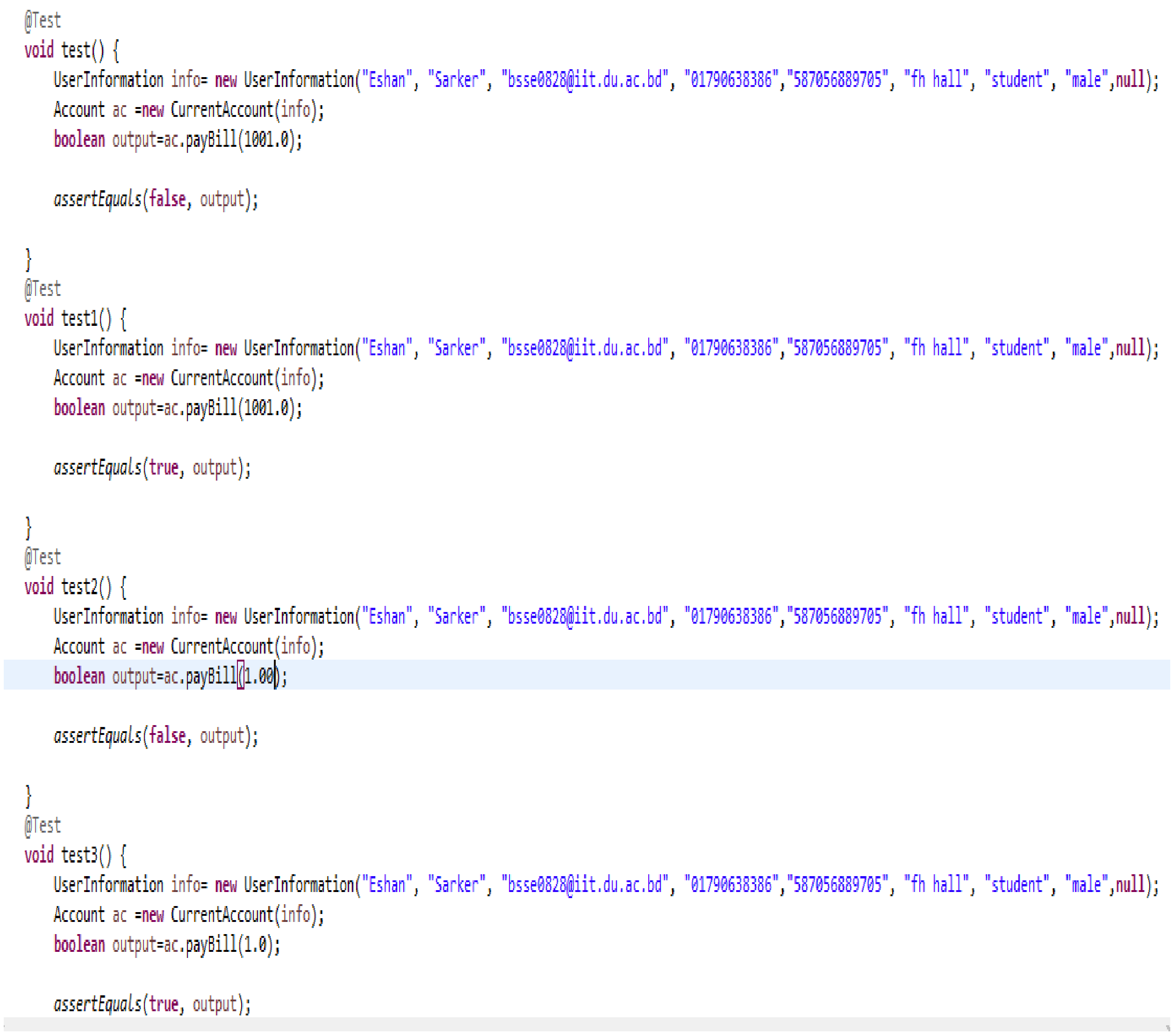
Test id	purpose	precondition	input	Expected output	Actual output	Test result

T26	To get PIN		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	75348	Generate random PIN	failed
T27	To get PIN		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	75348	Generate Random PIN	failed

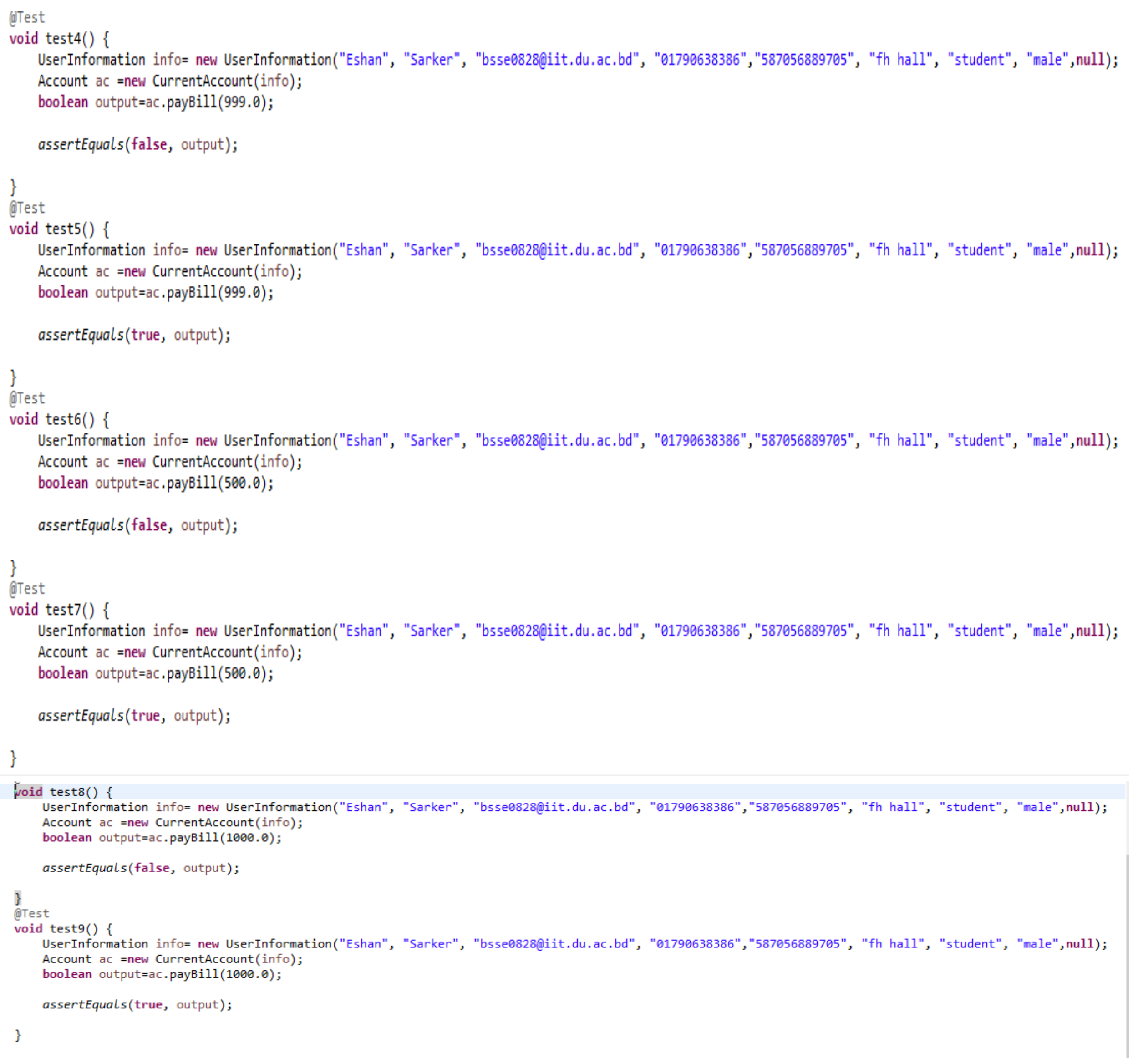
Table 20: Test Specification of getPIN

payBill method from CurrentAccountClass

TestCase:28,29,30,31,32,33,34,35,36



```java
@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.payBill(1001.0);

    assertEquals(false, output);

}
@Test
void test1() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.payBill(1001.0);

    assertEquals(true, output);

}
@Test
void test2() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.payBill(1.00);

    assertEquals(false, output);

}
@Test
void test3() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.payBill(1.0);

    assertEquals(true, output);
```



```
@Test
void test4() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.payBill(999.0);

    assertEquals(false, output);

}
@Test
void test5() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.payBill(999.0);

    assertEquals(true, output);

}
@Test
void test6() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.payBill(500.0);

    assertEquals(false, output);

}
@Test
void test7() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.payBill(500.0);

    assertEquals(true, output);

}
void test8() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.payBill(1000.0);

    assertEquals(false, output);

}
@Test
void test9() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    boolean output=ac.payBill(1000.0);

    assertEquals(true, output);

}
```

Figure 22: test case of T28,29,30,31,32,33,34,35,36,37

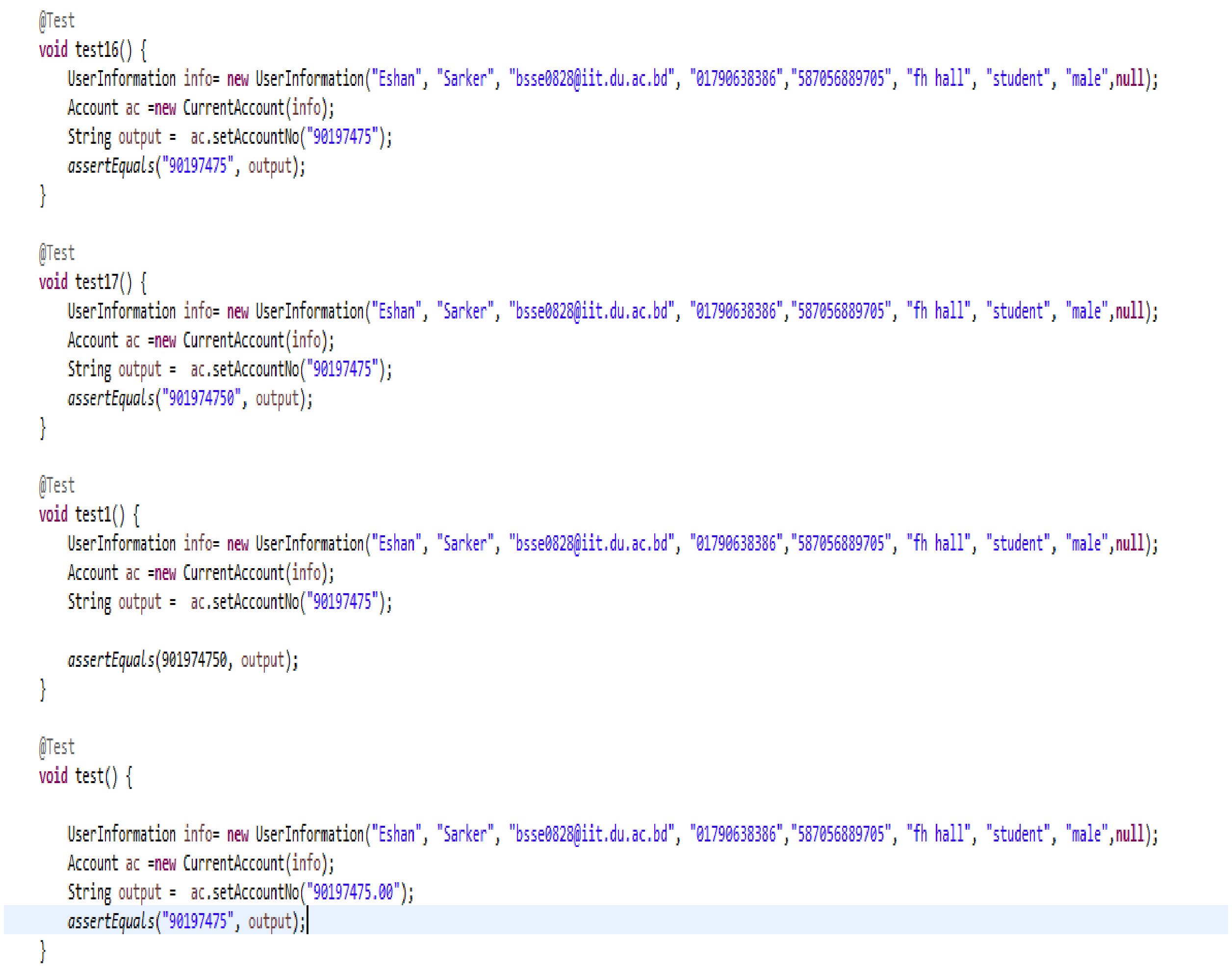
Test id	purpose	precond	input	Expected	Actual	Test

		ition		output	output	result
T28	To pay bill		1001.0	false	false	passed
T29	To pay bill		1001.0	true	false	failed
T30	To pay bill		1.0	false	false	passed
T31	To pay bill		1.0	true	false	failed
T32	To pay bill		999.0	false	false	passed
T33	To pay bill		999.0	true	false	failed
T34	To pay bill		500.00	false	false	passed
T35	To pay bill		500.00	true	false	failed
T36	To pay bill		1000.00	false	false	passed
T37			1000.0	true	false	failed

Table 22: Test Specification of payBill

setAccountNo() method from CurrentAccount class

TestCase: 38,39,40,41



```
@Test
void test16() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.setAccountNo("90197475");
    assertEquals("90197475", output);
}

@Test
void test17() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.setAccountNo("90197475");
    assertEquals("901974750", output);
}

@Test
void test1() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.setAccountNo("90197475");

    assertEquals(901974750, output);
}

@Test
void test() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.setAccountNo("90197475.00");
    assertEquals("90197475", output);
}
```

Figure 23: test case of T38,39,40,41

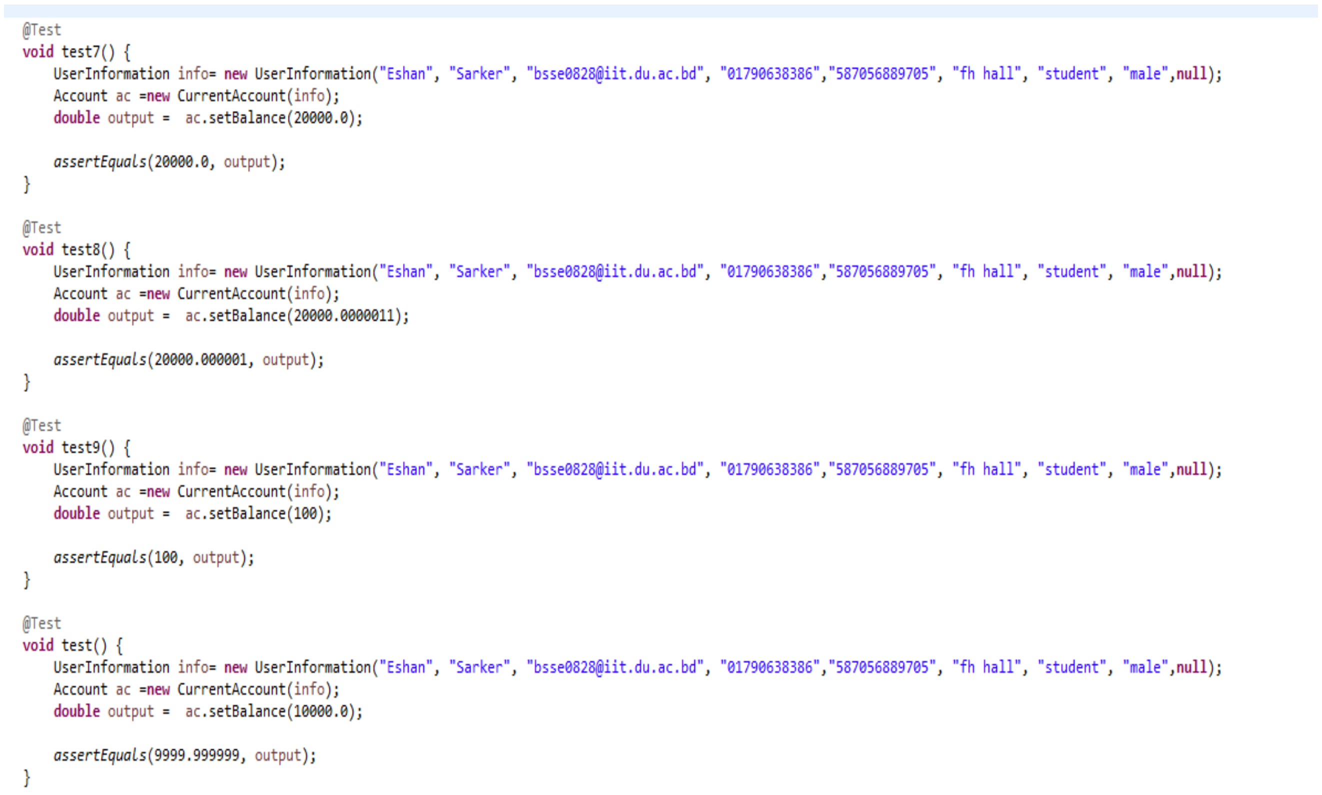
Test id	purpose	precondi	input	Expected	Actual	Test

		tion		output	output	result
T38	Set Account No		“90197475”	“90197475”	“90197475”	passed
T39	Set Account No		“90197475”	“90197470”	“90197475”	failed
T40	Set Account No		“90197475”	90197475	“90197475”	failed
T41	Set Account No		90197475.00	“90197475”	“90197475”	failed

Table 23: Test Specification of setAccountNo

SetBalance method for CurrentAccount Class

Testcase: 42,43,44,45



```
@Test
void test7() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output =  ac.setBalance(20000.0);

    assertEquals(20000.0, output);
}

@Test
void test8() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output =  ac.setBalance(20000.0000011);

    assertEquals(20000.000001, output);
}

@Test
void test9() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output =  ac.setBalance(100);

    assertEquals(100, output);
}

@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output =  ac.setBalance(10000.0);

    assertEquals(9999.999999, output);
}
```

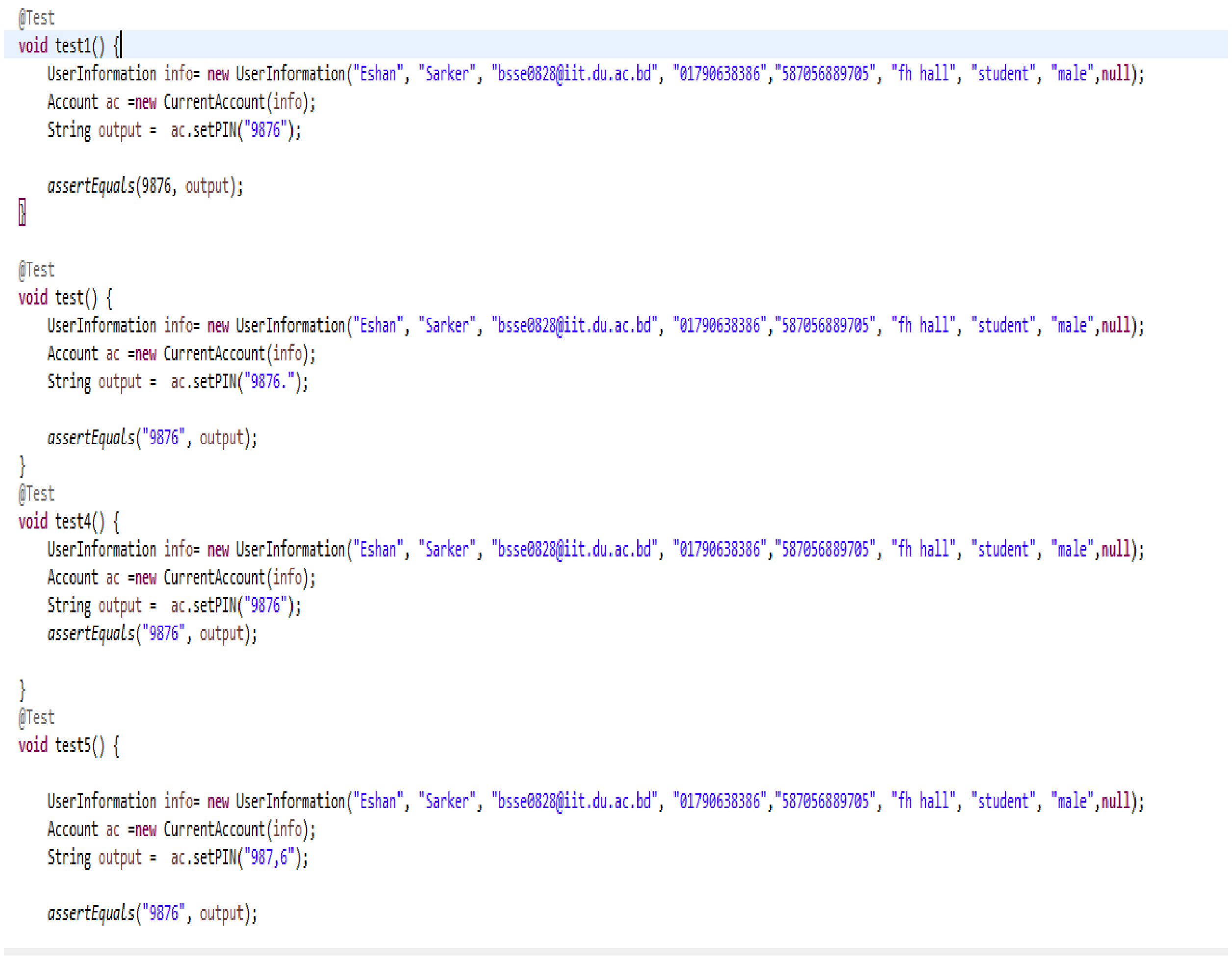
Figure 24: test case of T42,43,44,45

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T42	Set Account balance		2000.0	2000.0	2000.0	passed
T43	Set Account balance		2000.00011	2000.001	2000.00011	failed
T44	Set Account balance		100	100	100	passed
T45	Set Account balance		1000.0	999.999	1000.0	failed

Table 24: Test Specification of setBallance

setPIN method for CurrentAccount Class

Testcase: 46,47,48,49



```
@Test
void test1() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.setPIN("9876");

    assertEquals(9876, output);
}

@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.setPIN("9876.");

    assertEquals("9876", output);
}
@Test
void test4() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.setPIN("9876");
    assertEquals("9876", output);

}
@Test
void test5() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    String output =  ac.setPIN("987,6");

    assertEquals("9876", output);
```

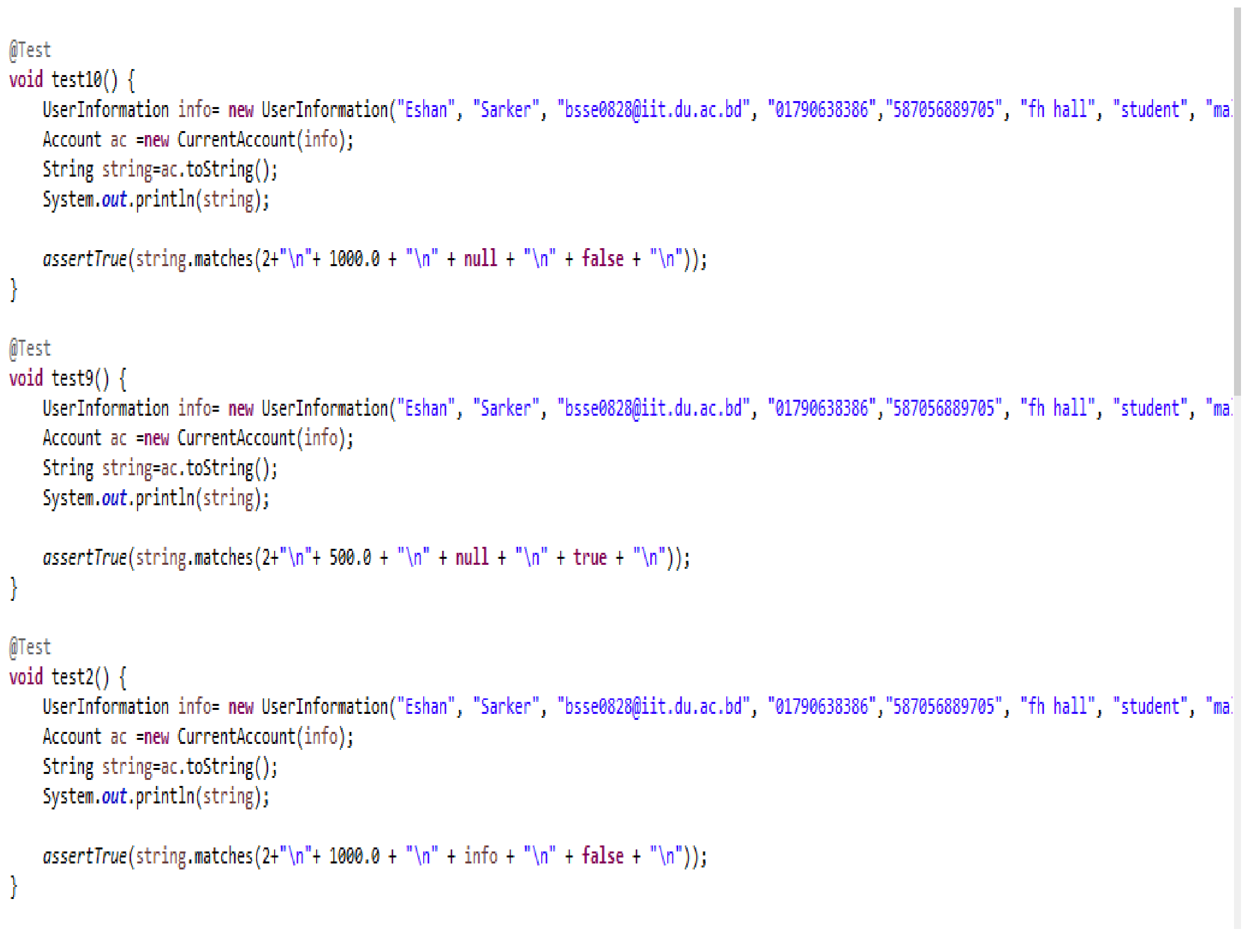
Figure 25: test case of T46,47,48,49

Test id	purpose	precondi tion	input	Expected output	Actual output	Test result
T46	Set PIN		“9876”	“9876”	“9876”	passed
T47	Set PIN		“9876,”	“9876”	“9876,”	failed
T48	Set PIN		“9876”	9876	“9876”	failed
T49	Set PIN		“987,6”	“9876”	“987,6”	failed

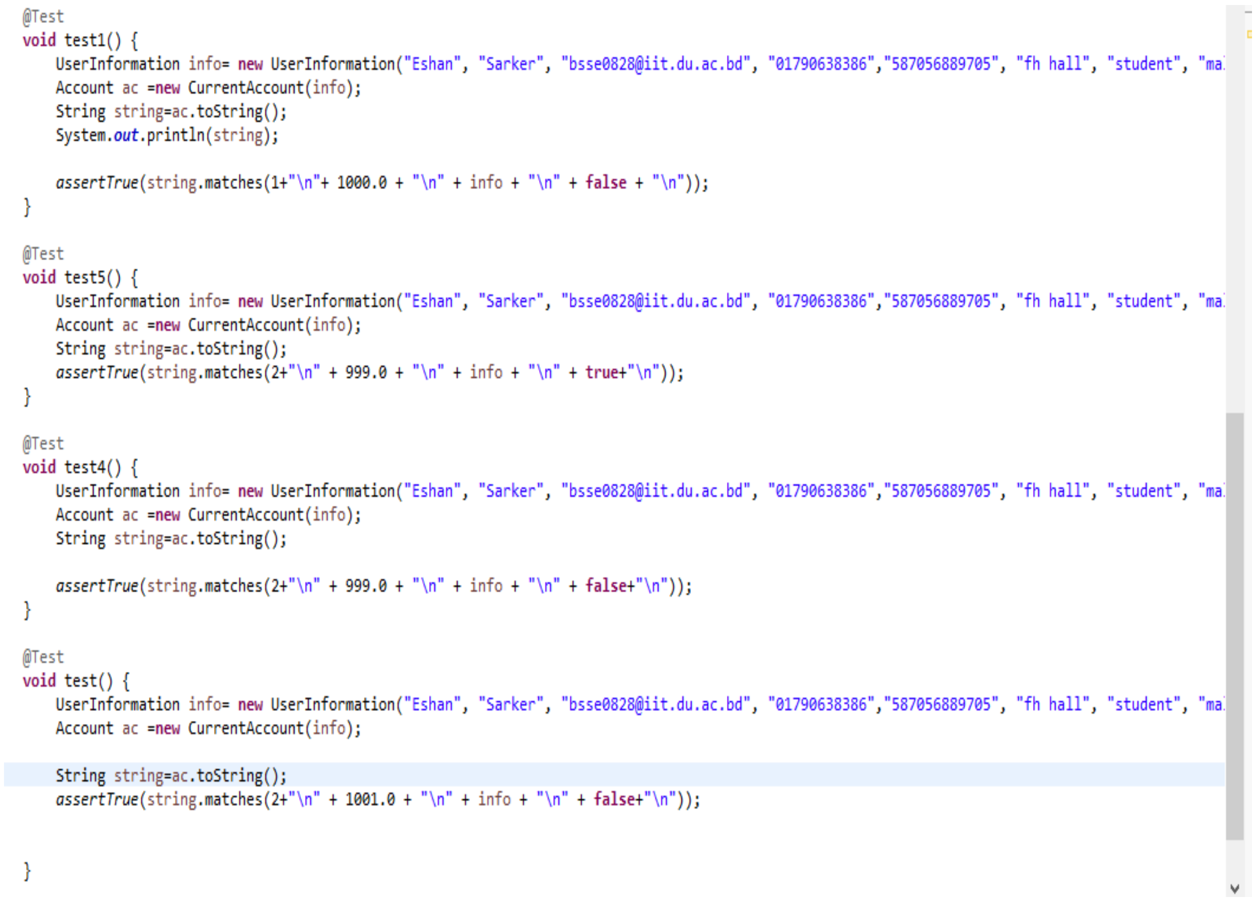
Table 25: Test Specification of setPIN

toString Testing from CurrentAccount Class

Testcase: 50,51,52,53,54,55,56



```java
@Test
void test10() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new CurrentAccount(info);
    String string=ac.toString();
    System.out.println(string);

    assertTrue(string.matches(2+"\n"+ 1000.0 + "\n" + null + "\n" + false + "\n"));
}

@Test
void test9() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new CurrentAccount(info);
    String string=ac.toString();
    System.out.println(string);

    assertTrue(string.matches(2+"\n"+ 500.0 + "\n" + null + "\n" + true + "\n"));
}

@Test
void test2() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new CurrentAccount(info);
    String string=ac.toString();
    System.out.println(string);

    assertTrue(string.matches(2+"\n"+ 1000.0 + "\n" + info + "\n" + false + "\n"));
}
```



```
@Test
void test1() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new CurrentAccount(info);
    String string=ac.toString();
    System.out.println(string);

    assertTrue(string.matches(1+"\n"+ 1000.0 + "\n" + info + "\n" + false + "\n"));
}

@Test
void test5() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new CurrentAccount(info);
    String string=ac.toString();
    assertTrue(string.matches(2+"\n" + 999.0 + "\n" + info + "\n" + true+"\n"));
}

@Test
void test4() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new CurrentAccount(info);
    String string=ac.toString();

    assertTrue(string.matches(2+"\n" + 999.0 + "\n" + info + "\n" + false+"\n"));
}

@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new CurrentAccount(info);

    String string=ac.toString();
    assertTrue(string.matches(2+"\n" + 1001.0 + "\n" + info + "\n" + false+"\n"));


}
```

Figure 26: test case of T50,51,52,53,54,55,56

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T50	Matching input and output		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student",	2 , 1000.0, null, false	2 , 1000.0, info, false	failed

			"male",null			
T51			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	2 , 500.0, null, true	2 , 1000.0, info, false	failed
T52	Matching input and output		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	2 , 1000.0, info, false	2 , 1000.0, info, false	passed
T53			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh	2 , 999.0, info, true	2 , 1000.0, info, false	failed

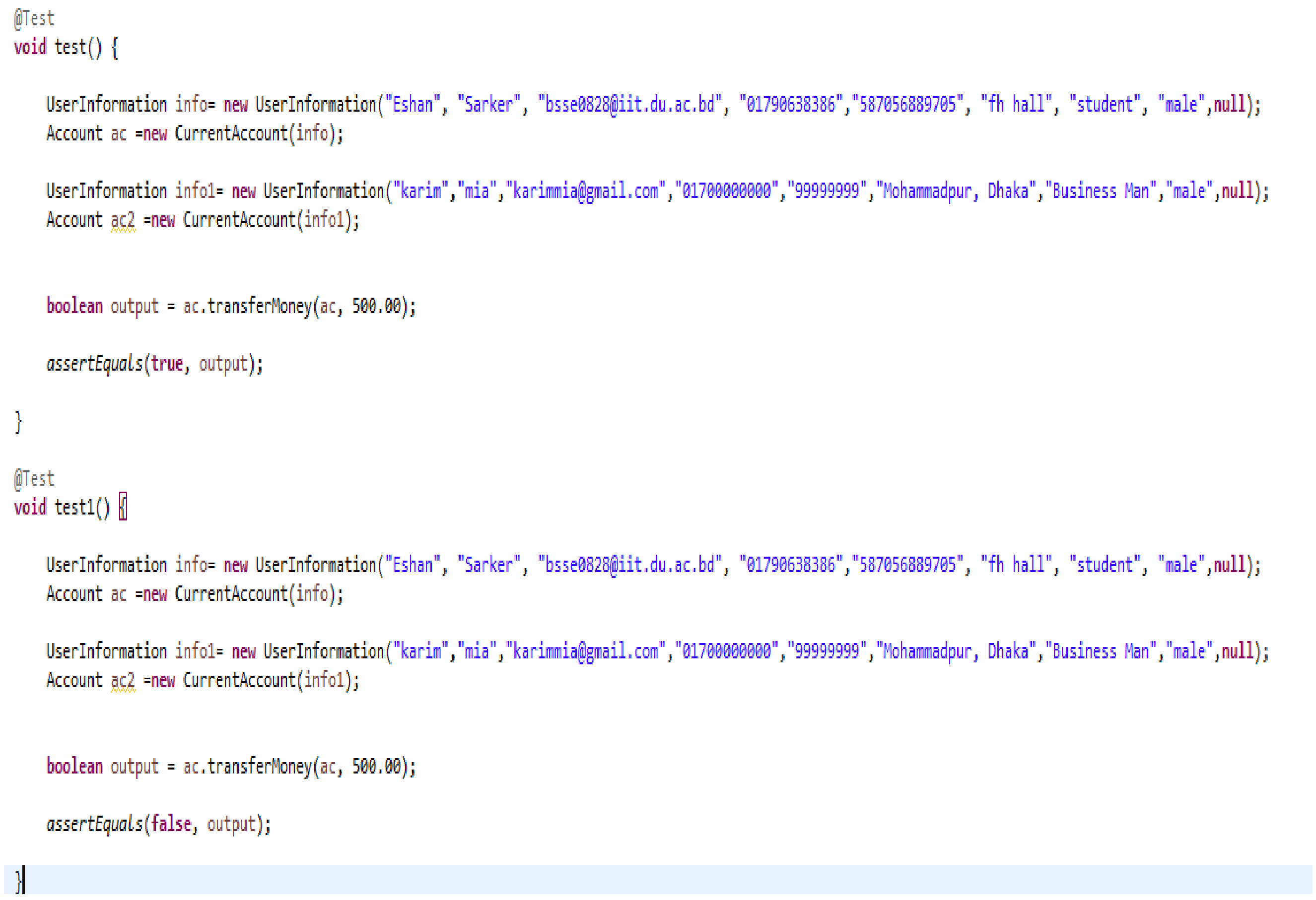
			hall", "student", "male",null			
T54			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	2 , 999.0, info, false	2 , 1000.0, info, false	failed
T55	Matching input and output		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	2 , 1001.0, info, true	2 , 1000.0, info, false	failed
T56			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","5	2 , 1001.0, info, false	2 , 1000.0, info, false	Failed

			87056889 705", "fh hall", "student", "male",null			

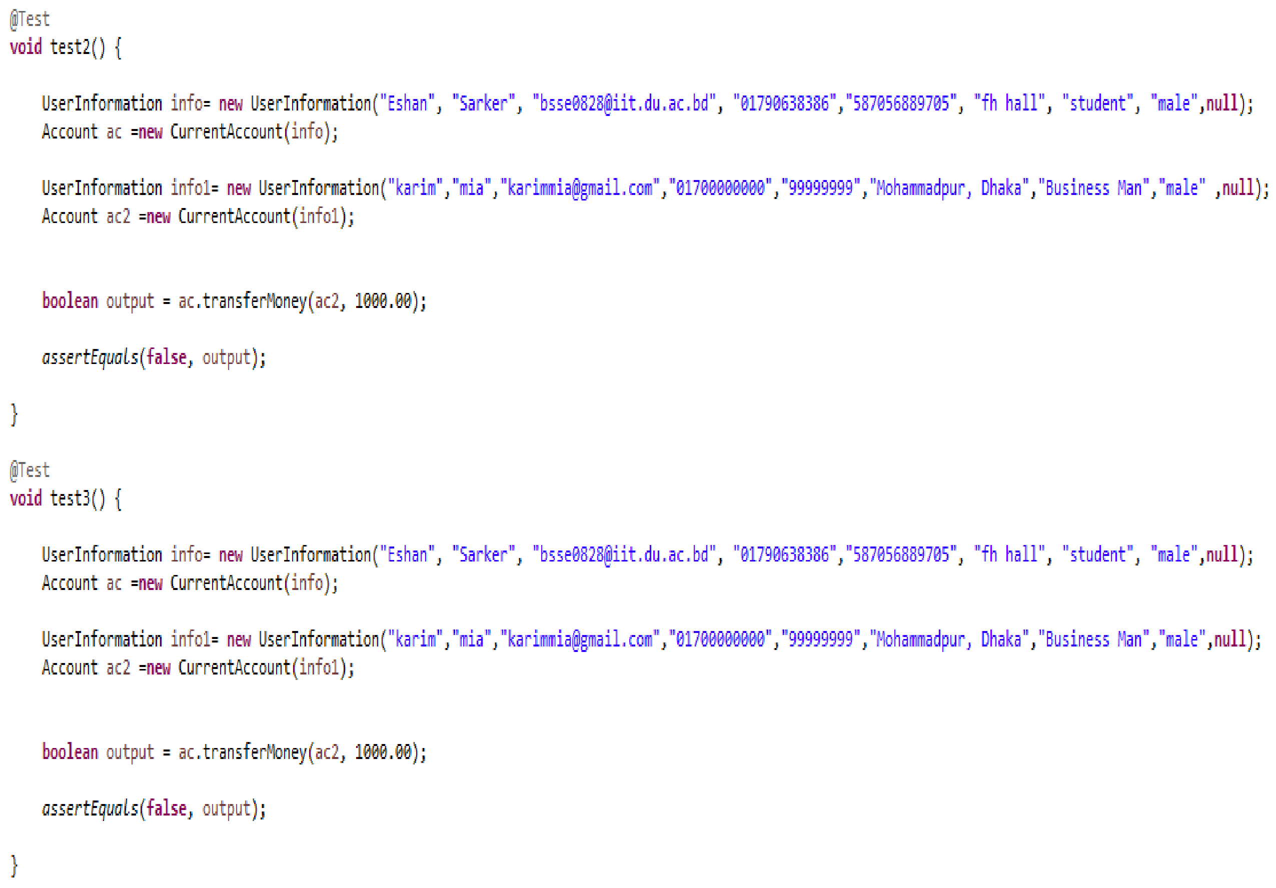
Table 26: Test Specification of toString

TransferMoney Method from CurrentAccount Class

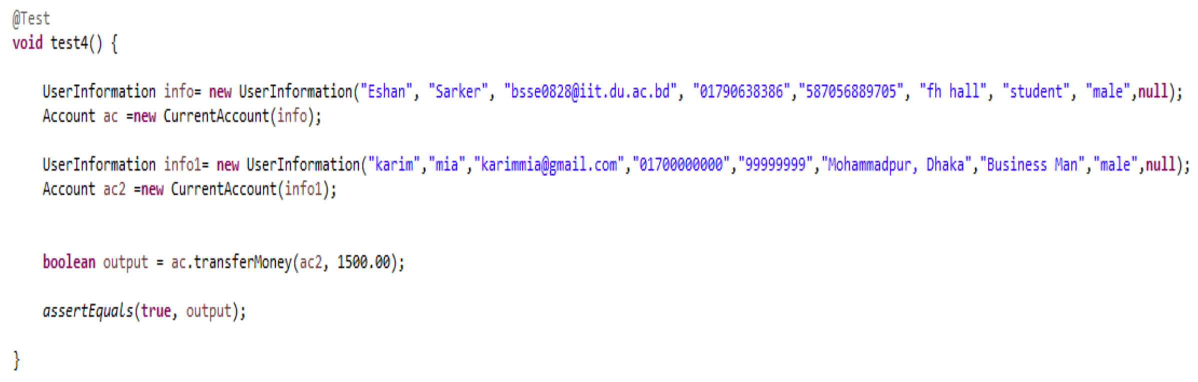
TestCase:57,58,59,60,61



```
@Test
void test() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);

    UserInformation info1= new UserInformation("karim","mia","karimmia@gmail.com","01700000000","99999999","Mohammadpur, Dhaka","Business Man","male",null);
    Account ac2 =new CurrentAccount(info1);


    boolean output = ac.transferMoney(ac, 500.00);

    assertEquals(true, output);

}

@Test
void test1() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);

    UserInformation info1= new UserInformation("karim","mia","karimmia@gmail.com","01700000000","99999999","Mohammadpur, Dhaka","Business Man","male",null);
    Account ac2 =new CurrentAccount(info1);


    boolean output = ac.transferMoney(ac, 500.00);

    assertEquals(false, output);

}
```



```
@Test
void test2() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);

    UserInformation info1= new UserInformation("karim","mia","karimmia@gmail.com","01700000000","99999999","Mohammadpur, Dhaka","Business Man","male" ,null);
    Account ac2 =new CurrentAccount(info1);


    boolean output = ac.transferMoney(ac2, 1000.00);

    assertEquals(false, output);

}

@Test
void test3() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);

    UserInformation info1= new UserInformation("karim","mia","karimmia@gmail.com","01700000000","99999999","Mohammadpur, Dhaka","Business Man","male",null);
    Account ac2 =new CurrentAccount(info1);


    boolean output = ac.transferMoney(ac2, 1000.00);

    assertEquals(false, output);

}
```



```
@Test
void test4() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);

    UserInformation info1= new UserInformation("karim","mia","karimmia@gmail.com","01700000000","99999999","Mohammadpur, Dhaka","Business Man","male",null);
    Account ac2 =new CurrentAccount(info1);


    boolean output = ac.transferMoney(ac2, 1500.00);

    assertEquals(true, output);

}
```

Figure 27: test case of T57,58,59,60,61

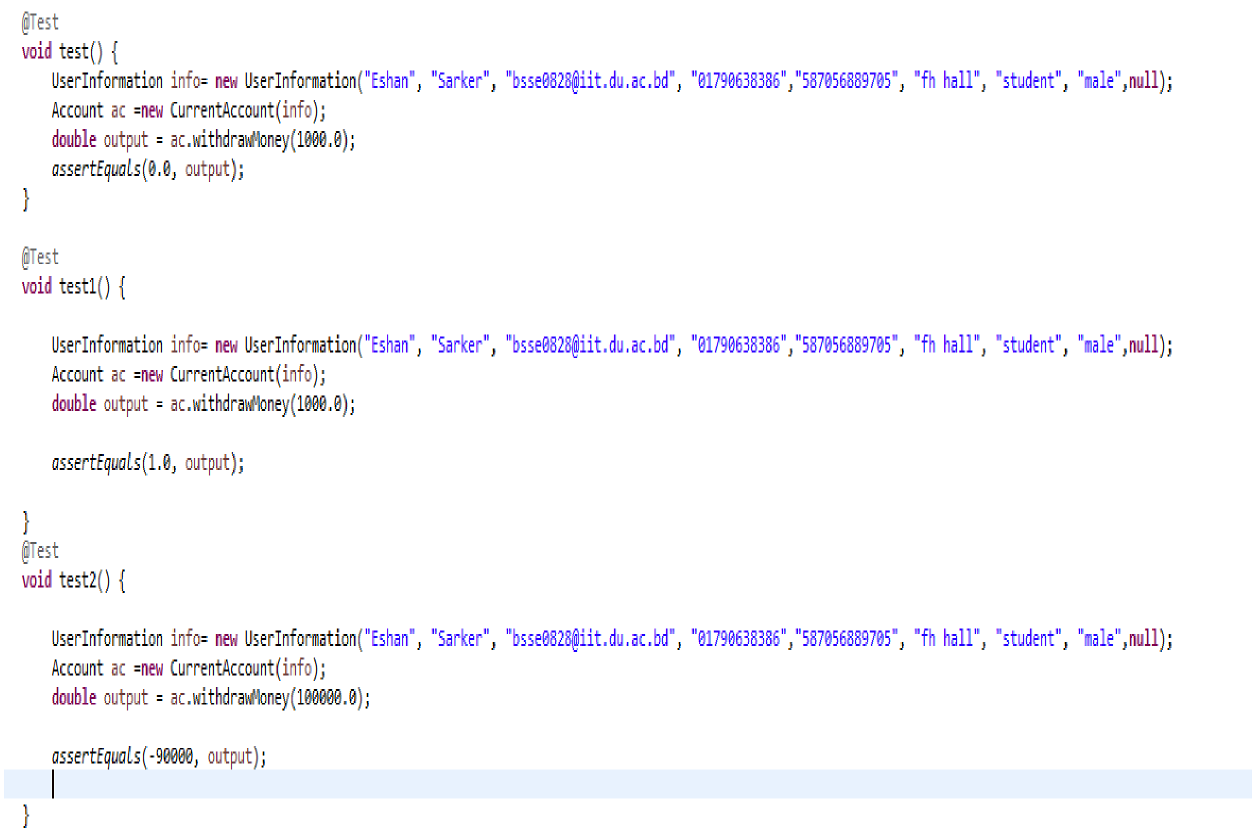
Test id	purpose	preconditi	input	Expected	Actual	Test

		on		output	output	result
T57	To check if tranfer money Is working		500.0	true	false	failed
T58			500.0	false	false	passed
T59			1000.0	false	false	passed
T60			1000.0	true	false	failed
T61			1500.0	true	false	failed

Table 27: Test Specification of transferMoney

Withdraw money method from CurrentAccount Class

Testcase: 62,63,64,65,66,67



```java
@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output = ac.withdrawMoney(1000.0);
    assertEquals(0.0, output);
}

@Test
void test1() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output = ac.withdrawMoney(1000.0);

    assertEquals(1.0, output);

}
@Test
void test2() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output = ac.withdrawMoney(100000.0);

    assertEquals(-90000, output);

}
```

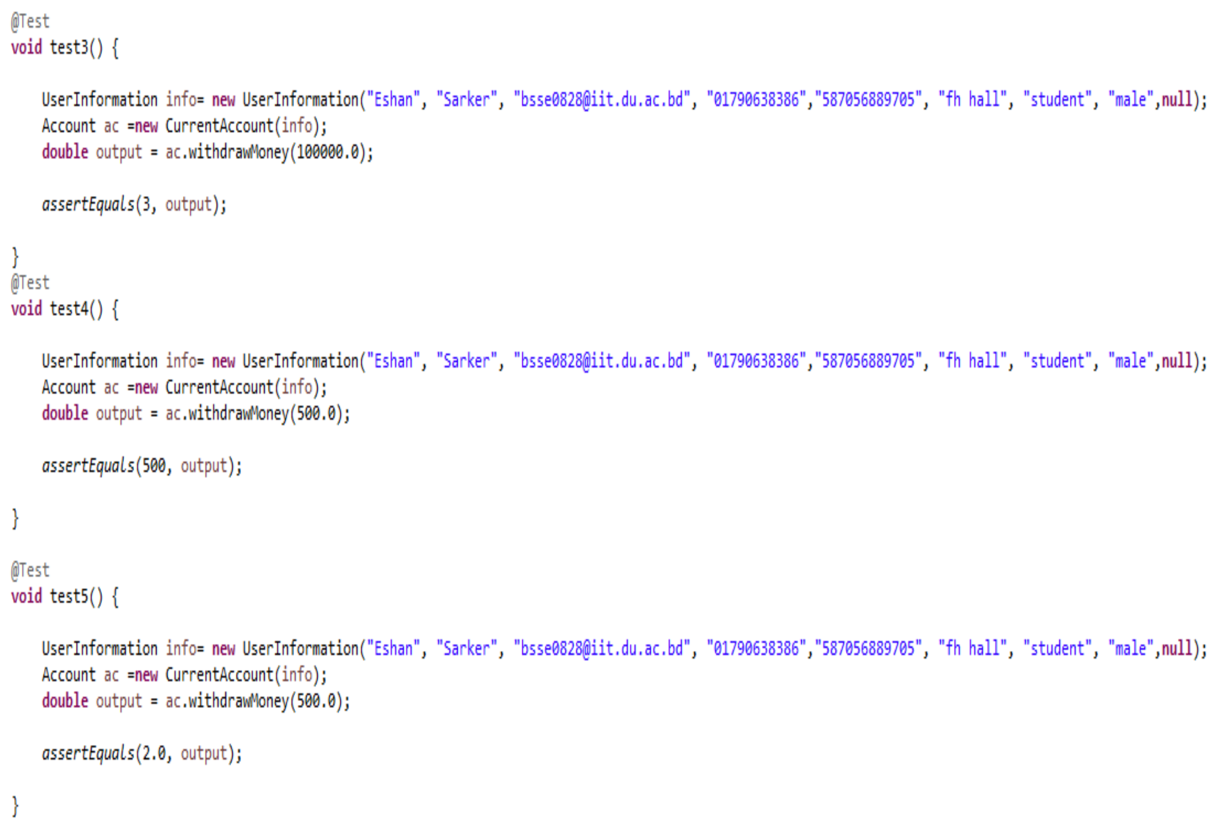

```
@Test
void test3() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output = ac.withdrawMoney(100000.0);

    assertEquals(3, output);

}
@Test
void test4() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output = ac.withdrawMoney(500.0);

    assertEquals(500, output);

}

@Test
void test5() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    Account ac =new CurrentAccount(info);
    double output = ac.withdrawMoney(500.0);

    assertEquals(2.0, output);

}
```

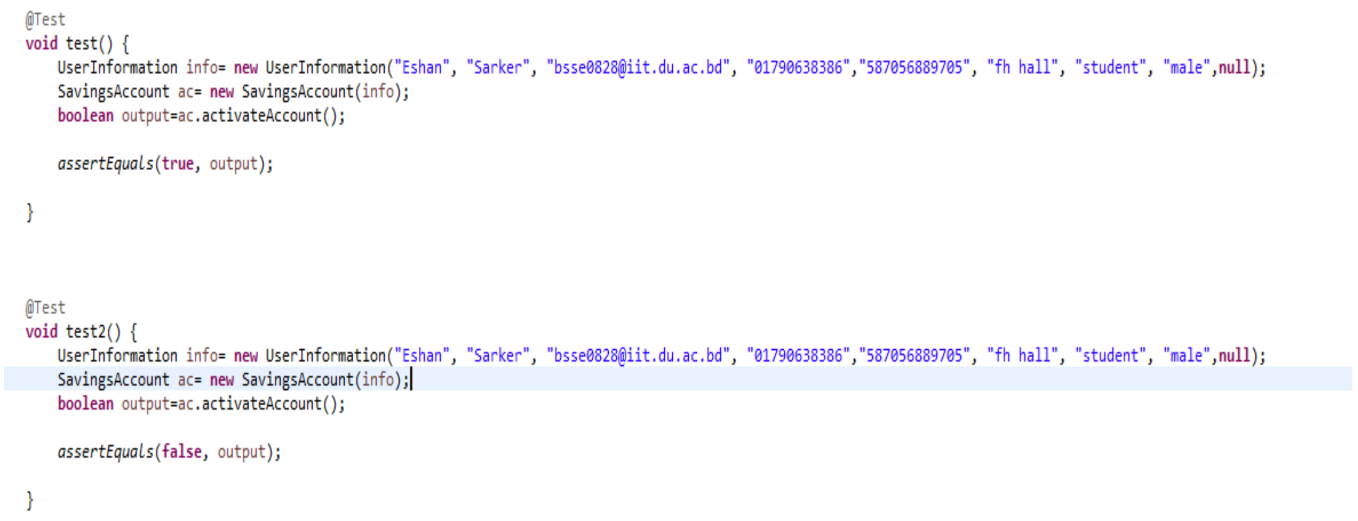
Figure 28: test case of T62,63,64,65,66,67

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T62	To check withdraw money		1000.0	0.0	1	failure
T63			1000.0	1	1	passed
T64			100000.0	-90000	3	failure
T65			100000.0	3	3	passed
T66			500.00	500	2	failure
T67			500.00	2	2	passed

Table 28: Test Specification of withdrawMoney

ActiveAccount method from SavingsAccount

Testcase: 68,69



```
@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.activateAccount();

    assertEquals(true, output);

}


@Test
void test2() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.activateAccount();

    assertEquals(false, output);

}
```

Figure 29: test case of T68,69

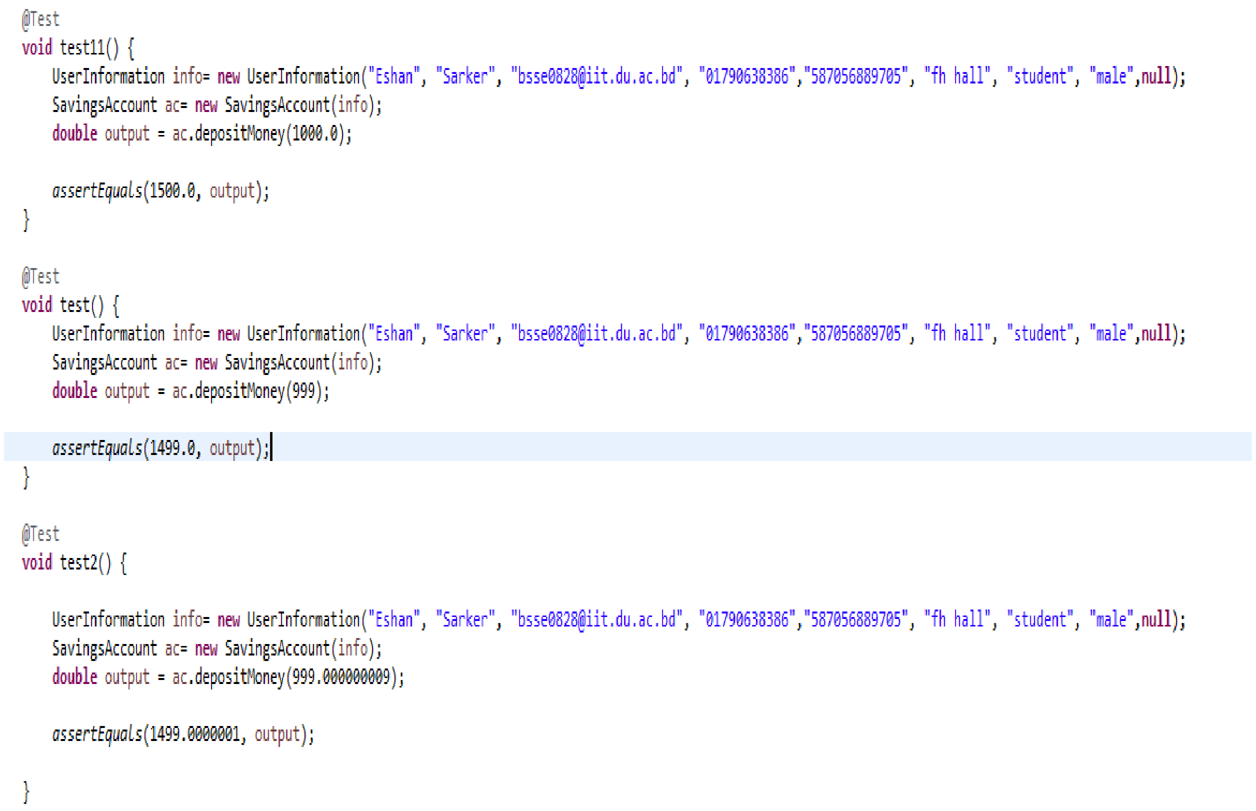
Test id	purpose	precondition	input	Expected output	Actual output	Test result
T68	To check if the account is active		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	true	true	passed
T69			"Eshan", "Sarker", "bsse082	false	true	failed

			8@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null			

Table 29: Test Specification of isActive

DepositMoney method from SavingsAccount

TestCase: 70,71,72



```
@Test
void test11() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    double output = ac.depositMoney(1000.0);

    assertEquals(1500.0, output);
}

@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    double output = ac.depositMoney(999);

    assertEquals(1499.0, output);
}

@Test
void test2() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    double output = ac.depositMoney(999.000000009);

    assertEquals(1499.0000001, output);

}
```

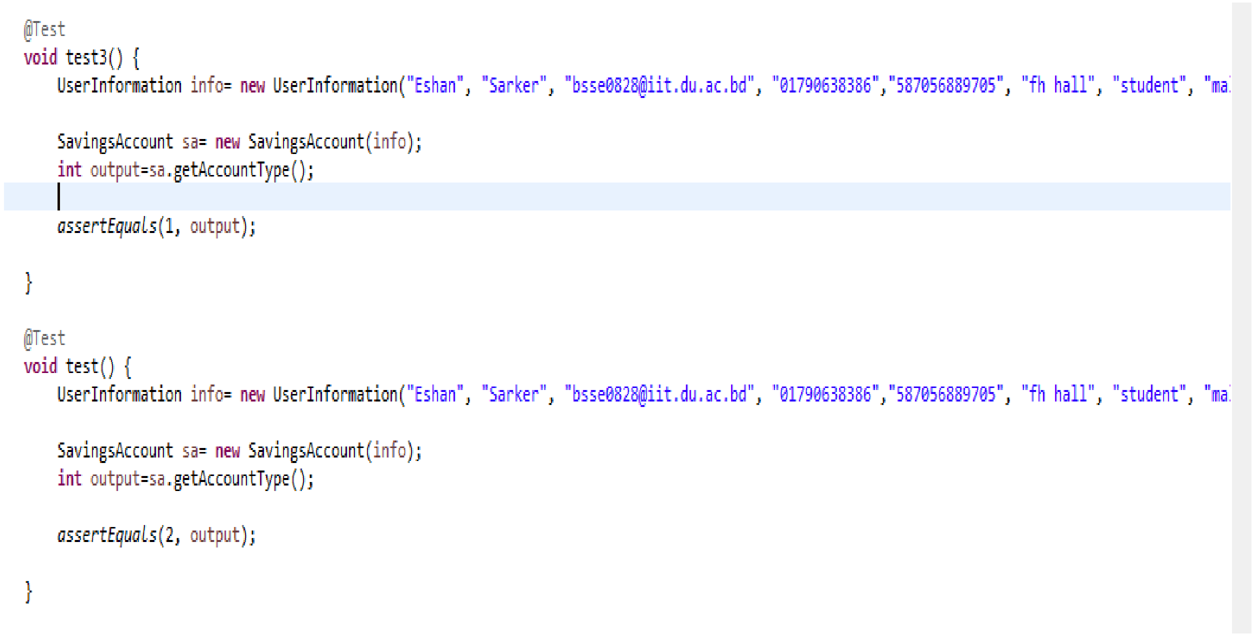
Figure 30: test case of T70,71,72

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T70	To deposit from savings account		1000.00	1500.0	1500.0	passed
T71			999	1499.0	1499.0	passed
T72			999.00000009	1499.0000001	1499.00000009	Failure

Table 30: Test Specification of Deposit(SavingsAccount)

getAccountType Method from SavingsAccount Class

TestCase:73,74



```
@Test
void test3() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma

    SavingsAccount sa= new SavingsAccount(info);
    int output=sa.getAccountType();

    assertEquals(1, output);

}

@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma

    SavingsAccount sa= new SavingsAccount(info);
    int output=sa.getAccountType();

    assertEquals(2, output);

}
```

Figure 31: test case of T73,74

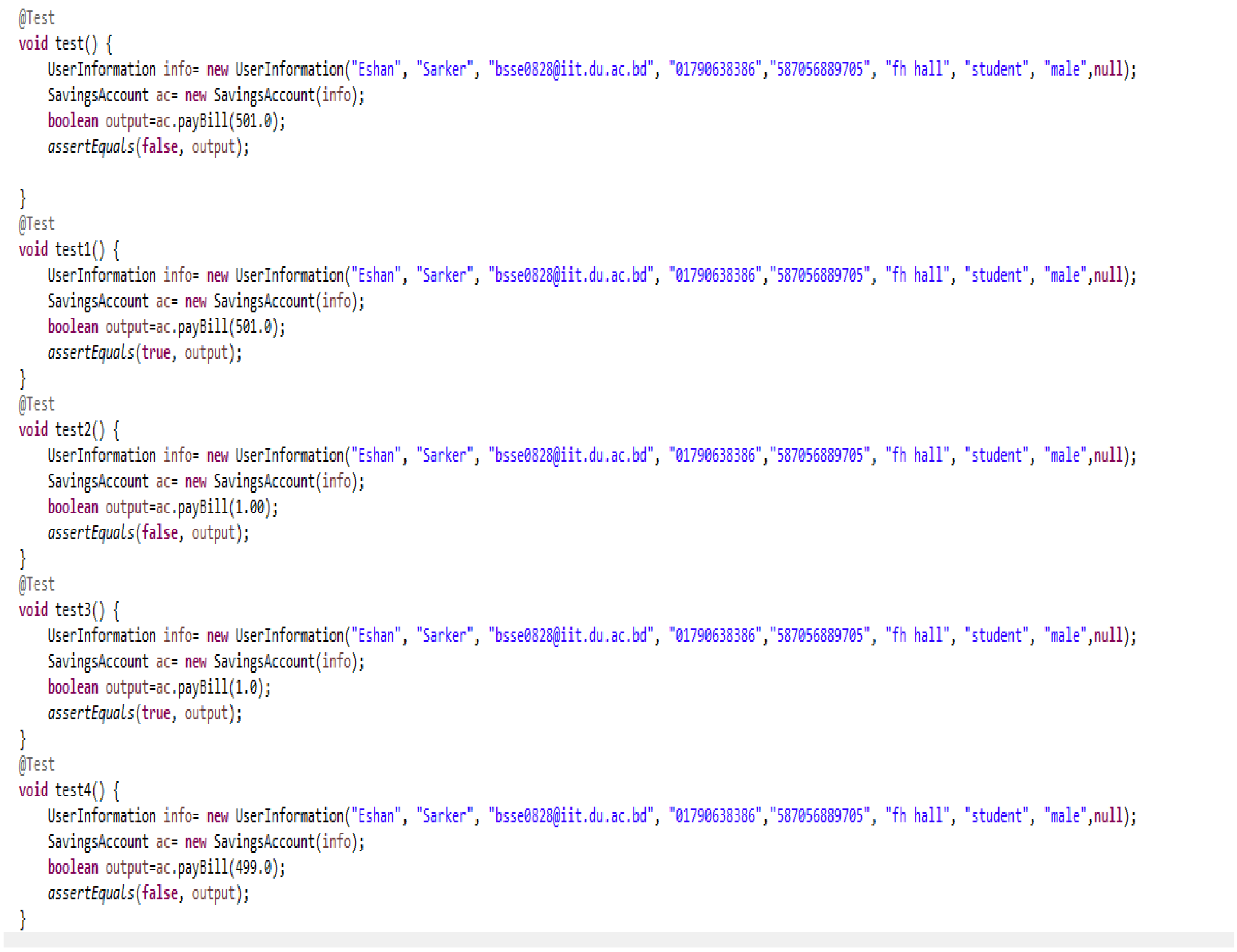
Test id	purpose	precondition	input	Expected output	Actual output	Test result
T73	To check account Type		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889	1	1	passed

			705", "fh hall", "student", "male",null			
T74			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	2	1	failed

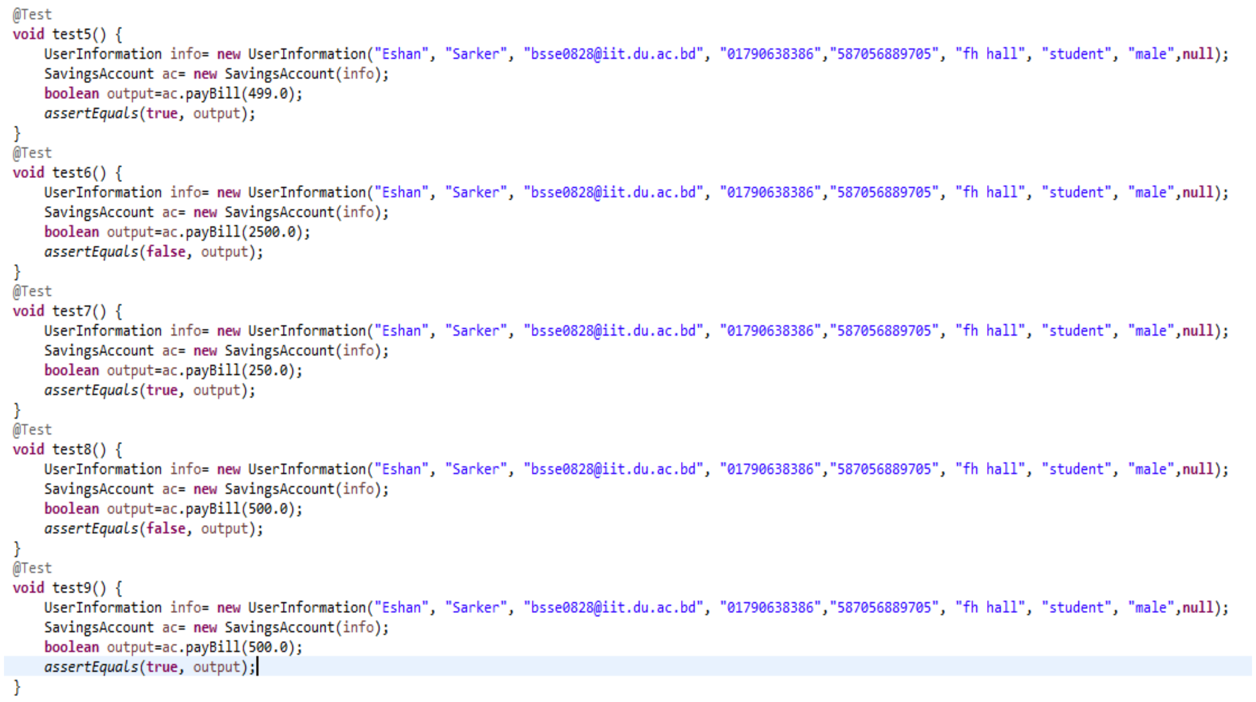
Table 31: Test Specification of AccountType(SavingsAccount)

PayBill method from SavingsAccount Class

Testcase:75,76,77,78,79,80,81,82,83,84



```java
@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.payBill(501.0);
    assertEquals(false, output);

}
@Test
void test1() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.payBill(501.0);
    assertEquals(true, output);
}
@Test
void test2() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.payBill(1.00);
    assertEquals(false, output);
}
@Test
void test3() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.payBill(1.0);
    assertEquals(true, output);
}
@Test
void test4() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.payBill(499.0);
    assertEquals(false, output);
}
```



```
@Test
void test5() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.payBill(499.0);
    assertEquals(true, output);
}
@Test
void test6() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.payBill(2500.0);
    assertEquals(false, output);
}
@Test
void test7() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.payBill(250.0);
    assertEquals(true, output);
}
@Test
void test8() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.payBill(500.0);
    assertEquals(false, output);
}
@Test
void test9() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    boolean output=ac.payBill(500.0);
    assertEquals(true, output);
}
```

Figure 32: test case of T75,76,77,78,79,80,81,82,83,84

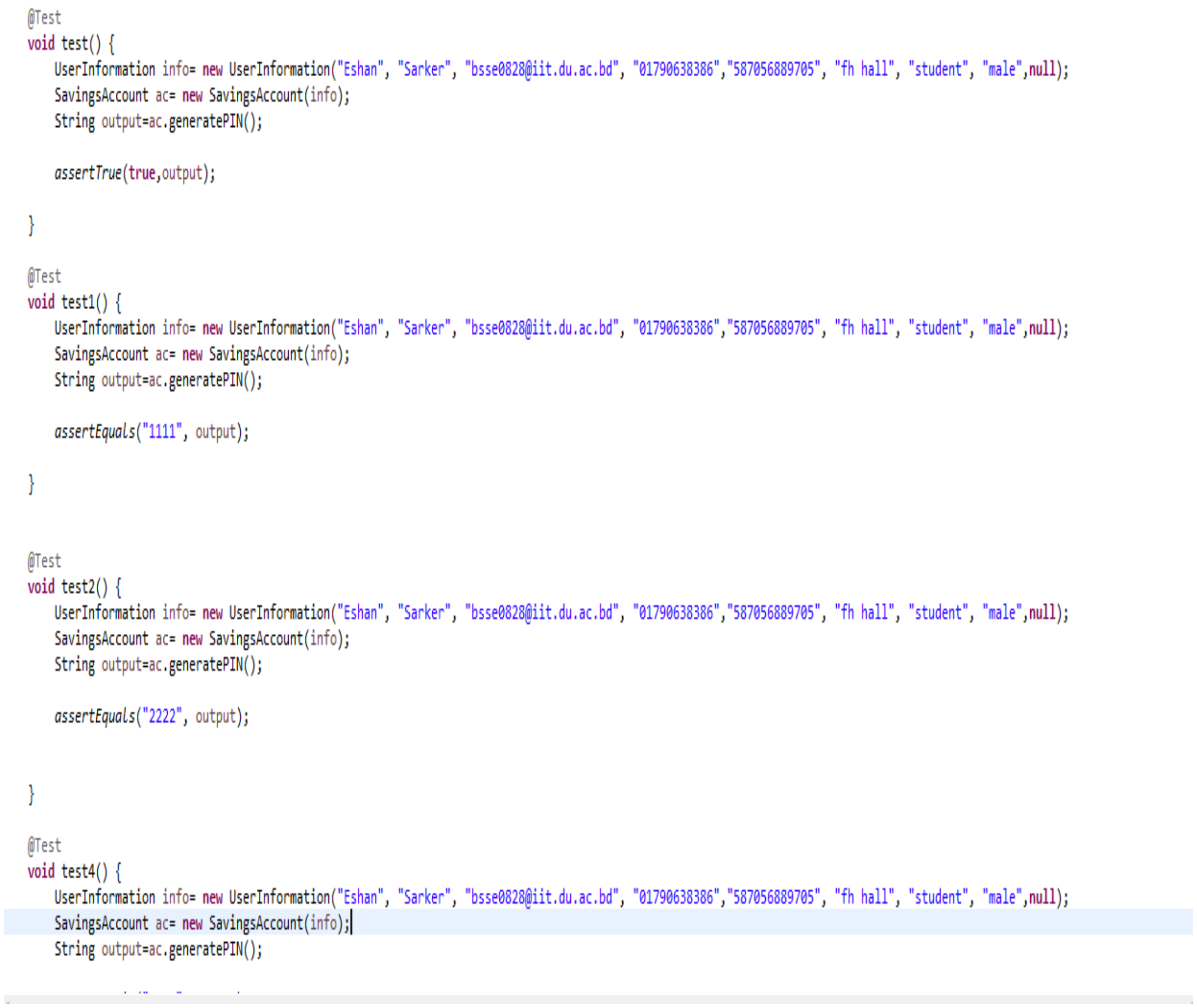
Test id	purpose	precondition	input	Expected output	Actual output	Test result
T75			501.0	false	false	passed
T76			501.0	true	false	failed
T77			1.0	false	false	passed
T78	Paybill check boolean result		1.0	true	false	failed
T79			499.0	false	false	passed
T80			499.00	true	false	failed
T81			250	false	false	passed

T82	Paybill check boolean result		250	true	false	failed
T83			500	false	false	passed
T84			500	true	false	failed

Table 32: Test Specification of payBill(SavingsAccount)

PINtesting method from SavingsAccount Class

TestCase:85,86,87,88



```
@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    String output=ac.generatePIN();

    assertTrue(true,output);

}

@Test
void test1() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    String output=ac.generatePIN();

    assertEquals("1111", output);

}


@Test
void test2() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    String output=ac.generatePIN();

    assertEquals("2222", output);


}

@Test
void test4() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    String output=ac.generatePIN();
```

Figure 33: test case of T85,86,87,88

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T85	Generate PIN		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","5	invalid	A random number	failed

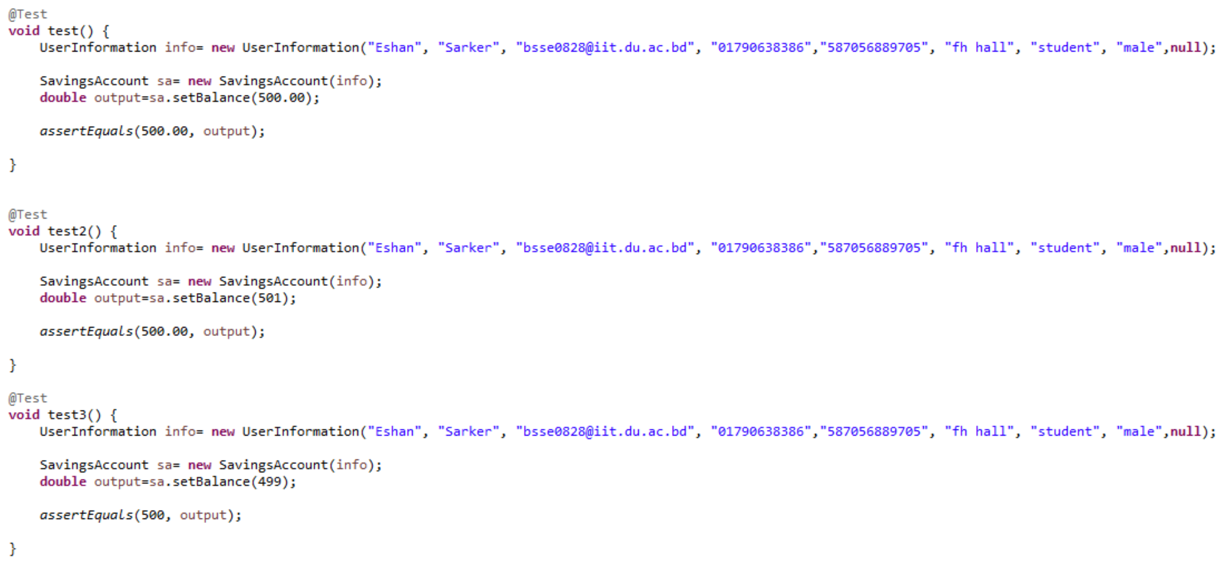
			87056889 705", "fh hall", "student", "male",null			
T86			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	invalid		failed
T87			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	invalid		failed
T88			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd",	invalid		Failed

			"01790638386","587056889705", "fh hall", "student", "male",null			

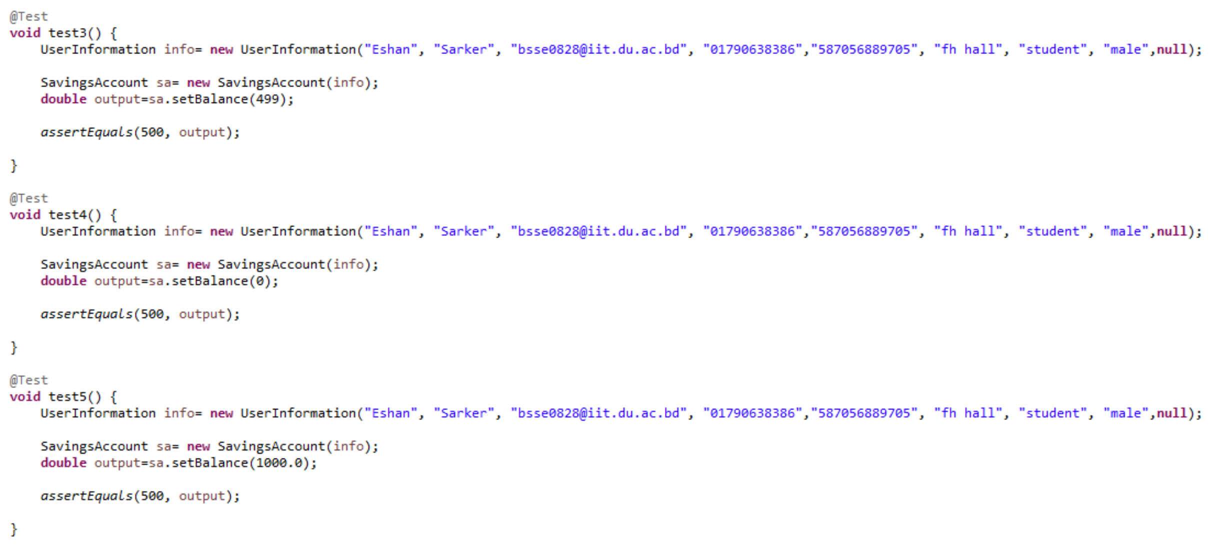
Table 33: Test Specification of payGenerate(SavingsAccount)

setBallance Testing from SavingsAccount Class

TestCase:89,90,91,92,93



```
@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    SavingsAccount sa= new SavingsAccount(info);
    double output=sa.setBalance(500.00);

    assertEquals(500.00, output);

}


@Test
void test2() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    SavingsAccount sa= new SavingsAccount(info);
    double output=sa.setBalance(501);

    assertEquals(500.00, output);

}

@Test
void test3() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    SavingsAccount sa= new SavingsAccount(info);
    double output=sa.setBalance(499);

    assertEquals(500, output);

}
```



```
@Test
void test3() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    SavingsAccount sa= new SavingsAccount(info);
    double output=sa.setBalance(499);

    assertEquals(500, output);

}

@Test
void test4() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    SavingsAccount sa= new SavingsAccount(info);
    double output=sa.setBalance(0);

    assertEquals(500, output);

}

@Test
void test5() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    SavingsAccount sa= new SavingsAccount(info);
    double output=sa.setBalance(1000.0);

    assertEquals(500, output);

}
```

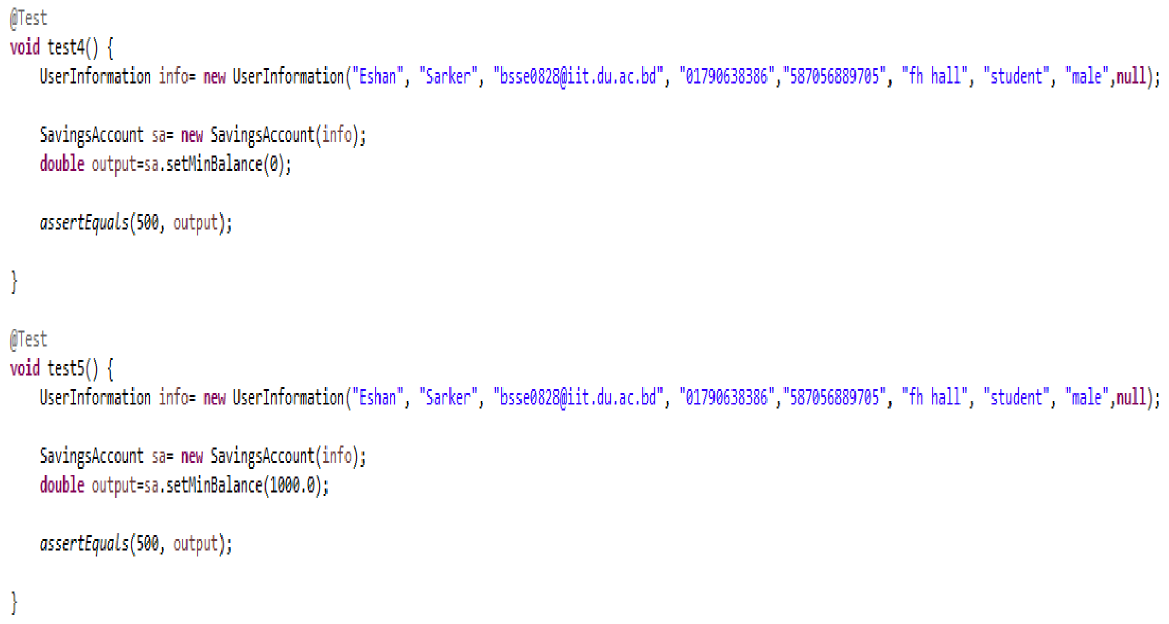
Figure 34: test case of T89,90,91,92,93

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T89	Set balance to current account		500.0	500.0	500.0	passed
T90			501.0	500.0	501.0	failed
T91			499.0	500.0	499.0	failed
T92	Paybill check boolean result		0.0	500.0	0.0	failed
T93			1000.0	500.0	1000.0	failed

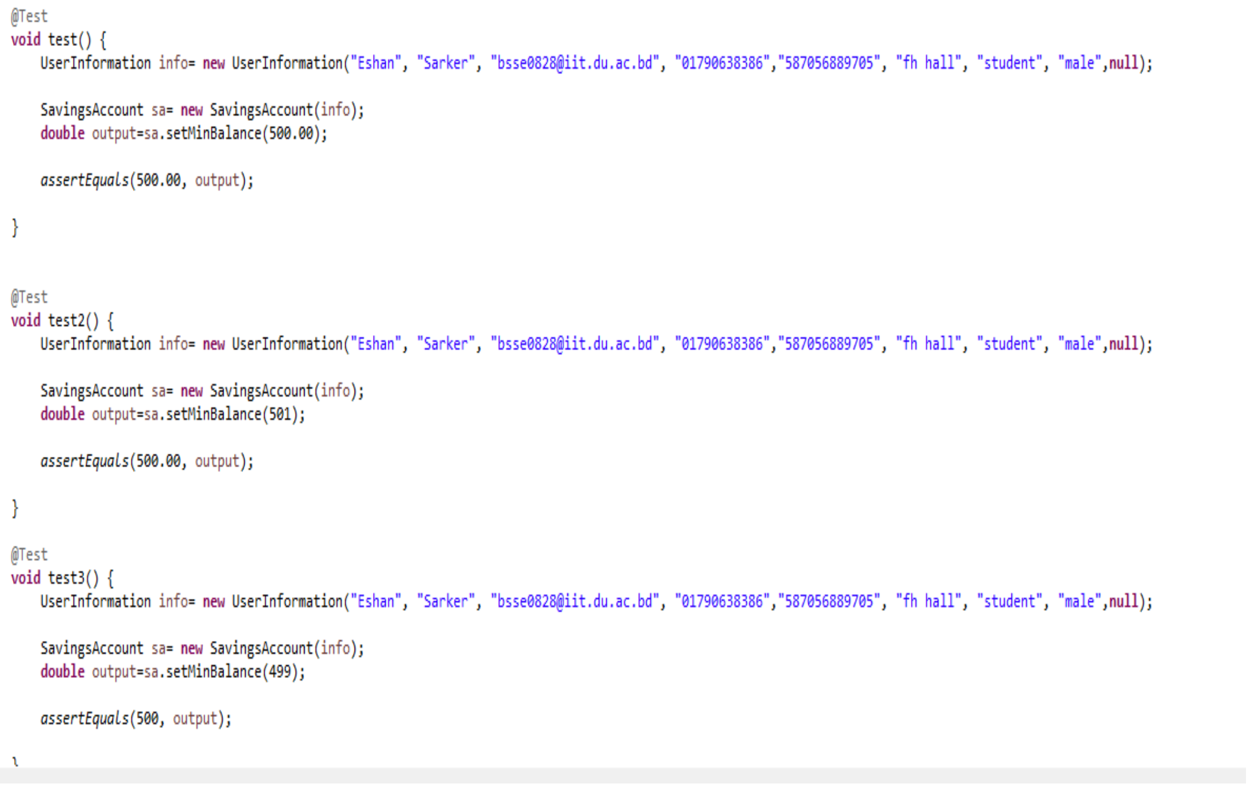
Table 34: Test Specification of SetBallance(SavingsAccount)

setMinBalance Method from SavingsAccount Class

Testcase: 94,95,96,97,96



```java
@Test
void test4() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    SavingsAccount sa= new SavingsAccount(info);
    double output=sa.setMinBalance(0);

    assertEquals(500, output);

}

@Test
void test5() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    SavingsAccount sa= new SavingsAccount(info);
    double output=sa.setMinBalance(1000.0);

    assertEquals(500, output);

}
```



```java
@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    SavingsAccount sa= new SavingsAccount(info);
    double output=sa.setMinBalance(500.00);

    assertEquals(500.00, output);

}

@Test
void test2() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    SavingsAccount sa= new SavingsAccount(info);
    double output=sa.setMinBalance(501);

    assertEquals(500.00, output);

}

@Test
void test3() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);

    SavingsAccount sa= new SavingsAccount(info);
    double output=sa.setMinBalance(499);

    assertEquals(500, output);

```

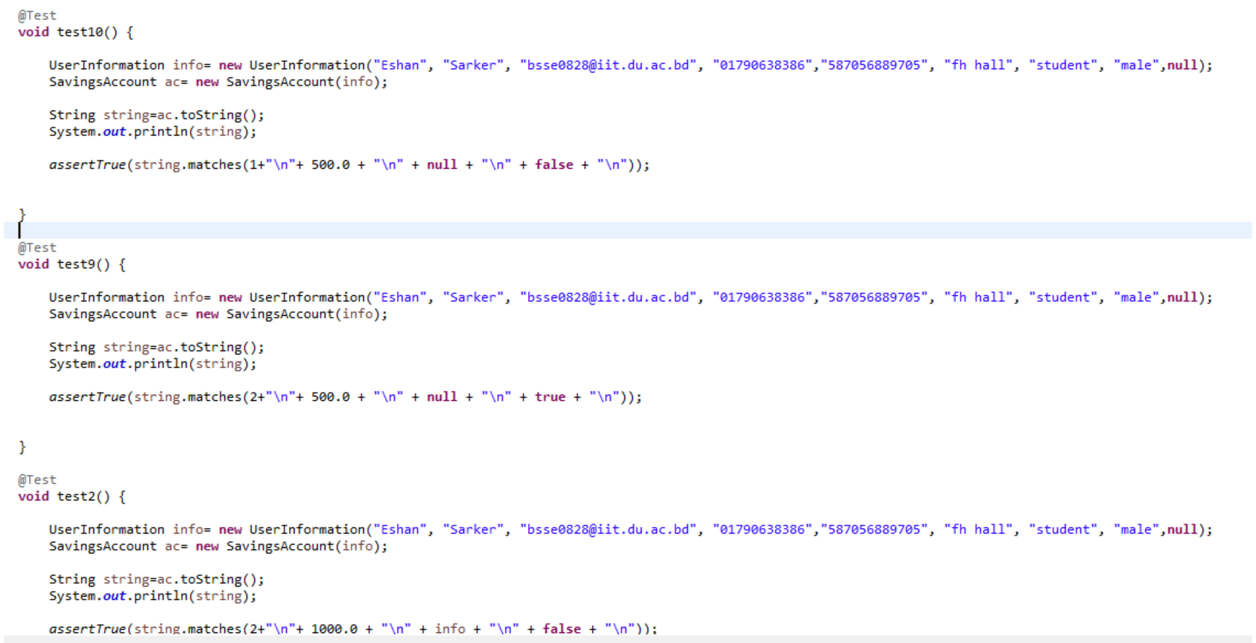
Figure 35: test case of T94,95,96,97,96

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T94	Set Min balance		500.0	500.0	500.0	passed
T95			501.0	500.0	501.0	failed
T96			499.0	500.0	499.0	failed
T97	Set Min balance		0.0	500.0	0.0	failed
T98			1000.0	500.0	1000.0	failed

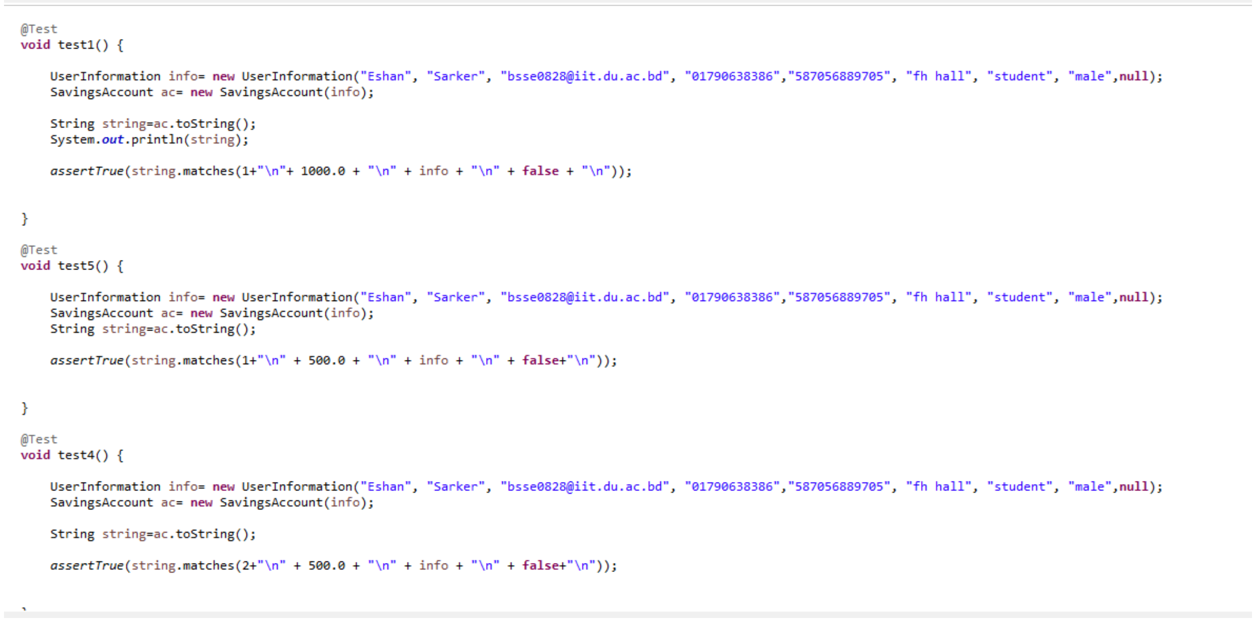
Table 35: Test Specification of SetMinBallance(SavingsAccount)

ToString Method from SavingsAccount Class

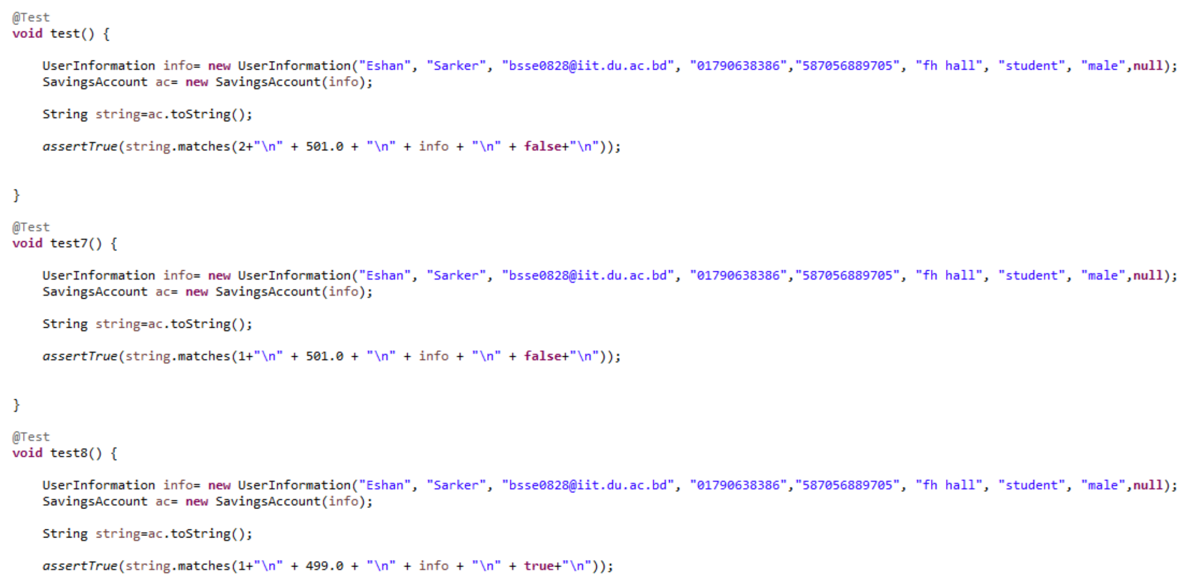
Testcase: 99,100,101,102,103,104,105,106,107



```java
@Test
void test10() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);

    String string=ac.toString();
    System.out.println(string);

    assertTrue(string.matches(1+"\n"+ 500.0 + "\n" + null + "\n" + false + "\n"));


}

@Test
void test9() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);

    String string=ac.toString();
    System.out.println(string);

    assertTrue(string.matches(2+"\n"+ 500.0 + "\n" + null + "\n" + true + "\n"));


}

@Test
void test2() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);

    String string=ac.toString();
    System.out.println(string);

    assertTrue(string.matches(2+"\n"+ 1000.0 + "\n" + info + "\n" + false + "\n"));
```



```java
@Test
void test1() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);

    String string=ac.toString();
    System.out.println(string);

    assertTrue(string.matches(1+"\n"+ 1000.0 + "\n" + info + "\n" + false + "\n"));


}
@Test
void test5() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    String string=ac.toString();

    assertTrue(string.matches(1+"\n" + 500.0 + "\n" + info + "\n" + false+"\n"));


}
@Test
void test4() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);

    String string=ac.toString();

    assertTrue(string.matches(2+"\n" + 500.0 + "\n" + info + "\n" + false+"\n"));
```



```java
@Test
void test() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);

    String string=ac.toString();

    assertTrue(string.matches(2+"\n" + 501.0 + "\n" + info + "\n" + false+"\n"));


}
@Test
void test7() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);

    String string=ac.toString();

    assertTrue(string.matches(1+"\n" + 501.0 + "\n" + info + "\n" + false+"\n"));


}
@Test
void test8() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);

    String string=ac.toString();

    assertTrue(string.matches(1+"\n" + 499.0 + "\n" + info + "\n" + true+"\n"));
```

Figure 36: test case of T99,100,101,102,103,104,105,106,107

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T99	Test status of account		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	1, 500, null, false	1, 500, info, false	failed
T100			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd",	2, 500, null, true	1, 500, info, false	failed

			"01790638386","587056889705", "fh hall", "student", "male",null			
T101			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	2, 1000, info, false	1, 500, info, false	failed
T102	Test matching status of account		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	1, 1000, info, false	1, 500, info, false	passed
T103			"Eshan", "Sarker", "bsse082	1, 500, info,	1, 500, info,	failed

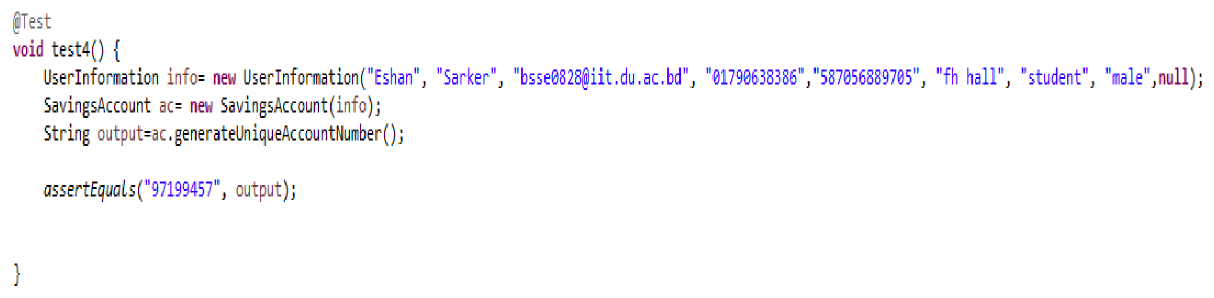
			8@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	false	false	
T104			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	2, 500, info, false	1, 500, info, false	failed
T105	Test matching status of account		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	2, 501, info, false	1, 500, info, false	failed
T106			"Eshan",	2,	1,	failed

			"Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	499, info, false	500, info, false	
T107	Test matching status of account		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	1, 499, info, false	1, 500, info, false	failed

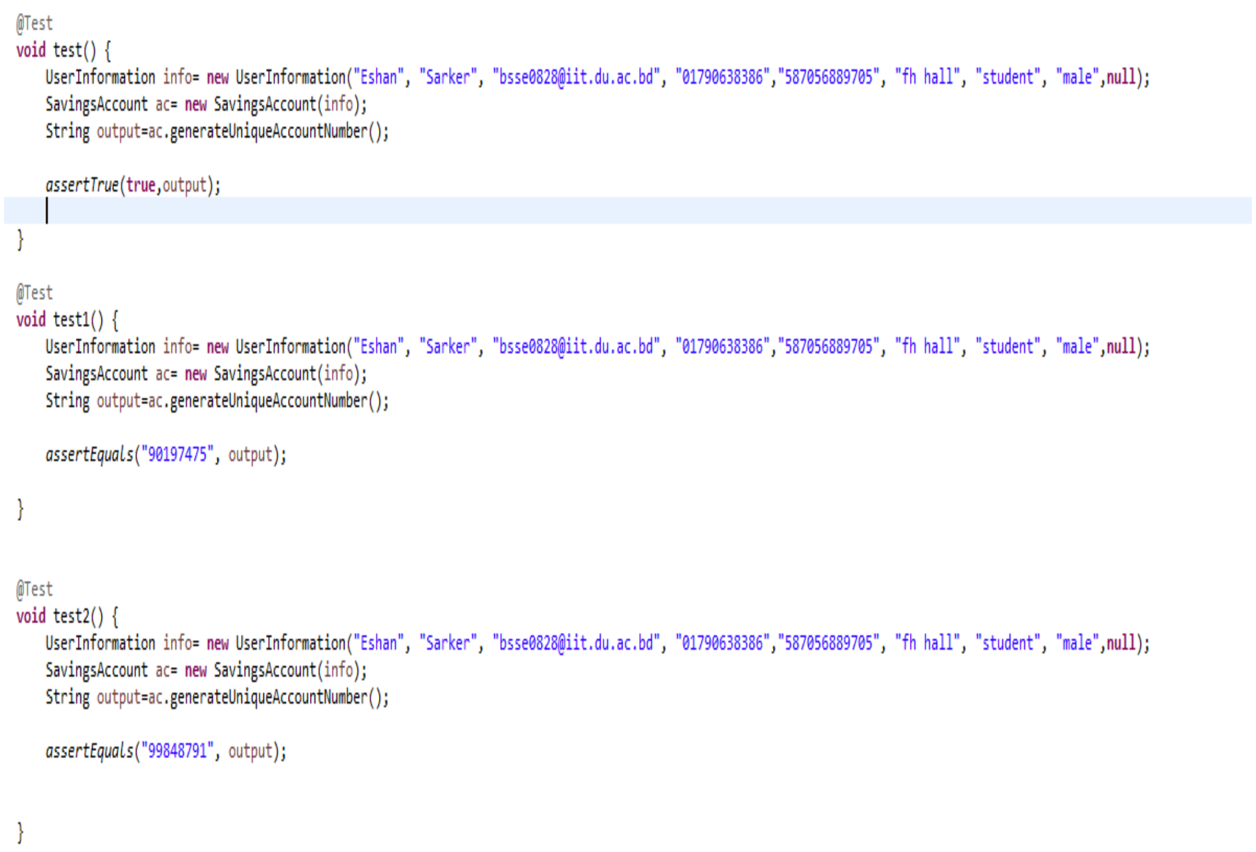
Table 36: Test Specification of toString(SavingsAccount)

uniqueAccount() Method from SavingsAccount Class

Testcase:108,109,110,111



```
@Test
void test4() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    String output=ac.generateUniqueAccountNumber();

    assertEquals("97199457", output);


}
```



```
@Test
void test() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    String output=ac.generateUniqueAccountNumber();

    assertTrue(true,output);
    
}

@Test
void test1() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    String output=ac.generateUniqueAccountNumber();

    assertEquals("90197475", output);

}


@Test
void test2() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    String output=ac.generateUniqueAccountNumber();

    assertEquals("99848791", output);


}
```

Figure 37: test case of T108,109,110,111

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T108	Generate Unique		"Eshan", "Sarker",	true	true	passed

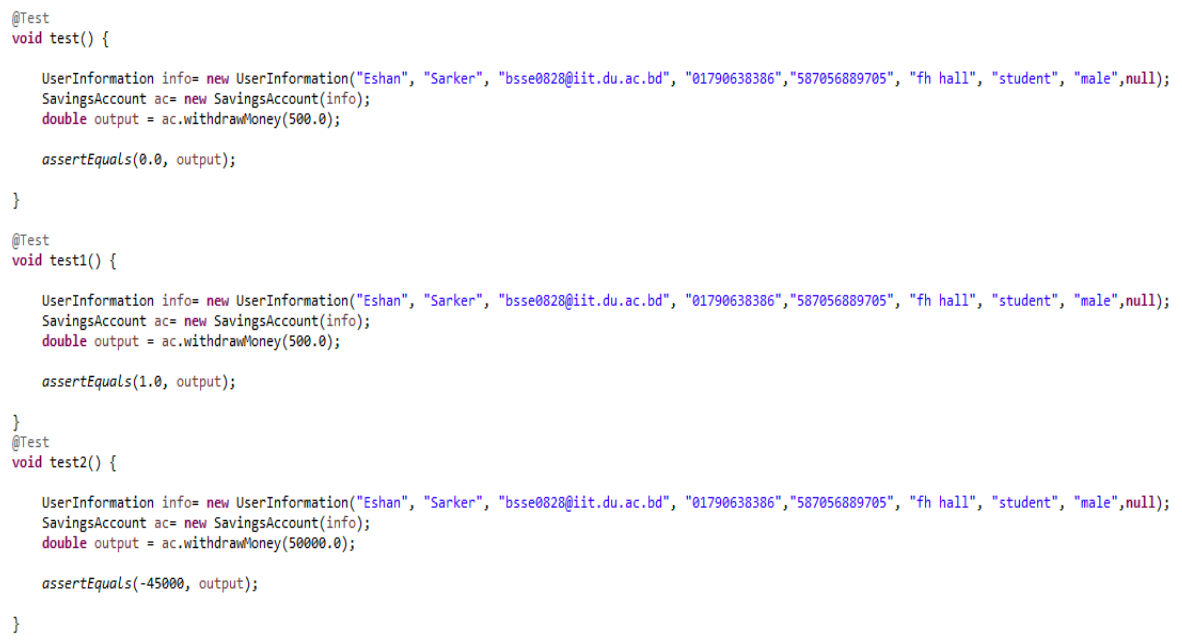
	Account number		"bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null			
T109			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	invalid	Random number	failed
T110			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	invalid		failed

T111	Generate Unique Account number		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	invalid		failed

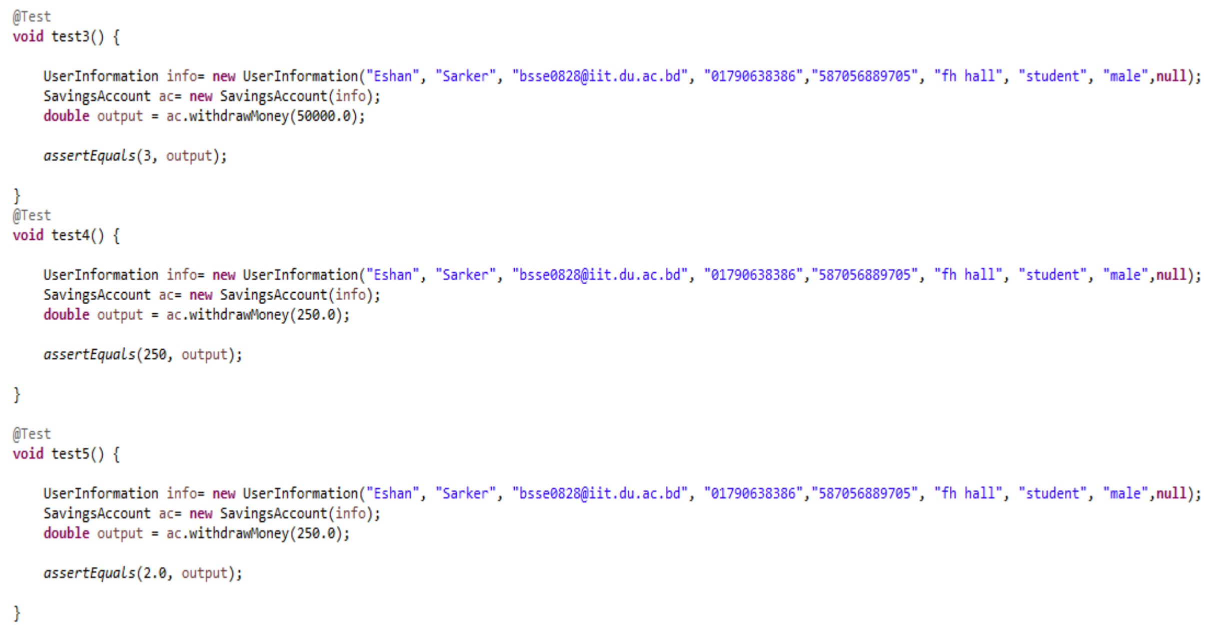
Table 37: Test Specification of generateUniquePIN(SavingsAccount)

Withdraw method from SavingsAccount Class

Testcase:112,113,114,115,116,117



```
@Test
void test() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    double output = ac.withdrawMoney(500.0);

    assertEquals(0.0, output);

}

@Test
void test1() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    double output = ac.withdrawMoney(500.0);

    assertEquals(1.0, output);

}
@Test
void test2() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    double output = ac.withdrawMoney(50000.0);

    assertEquals(-45000, output);

}
```



```
@Test
void test3() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    double output = ac.withdrawMoney(50000.0);

    assertEquals(3, output);

}
@Test
void test4() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    double output = ac.withdrawMoney(250.0);

    assertEquals(250, output);

}

@Test
void test5() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null);
    SavingsAccount ac= new SavingsAccount(info);
    double output = ac.withdrawMoney(250.0);

    assertEquals(2.0, output);

}
```

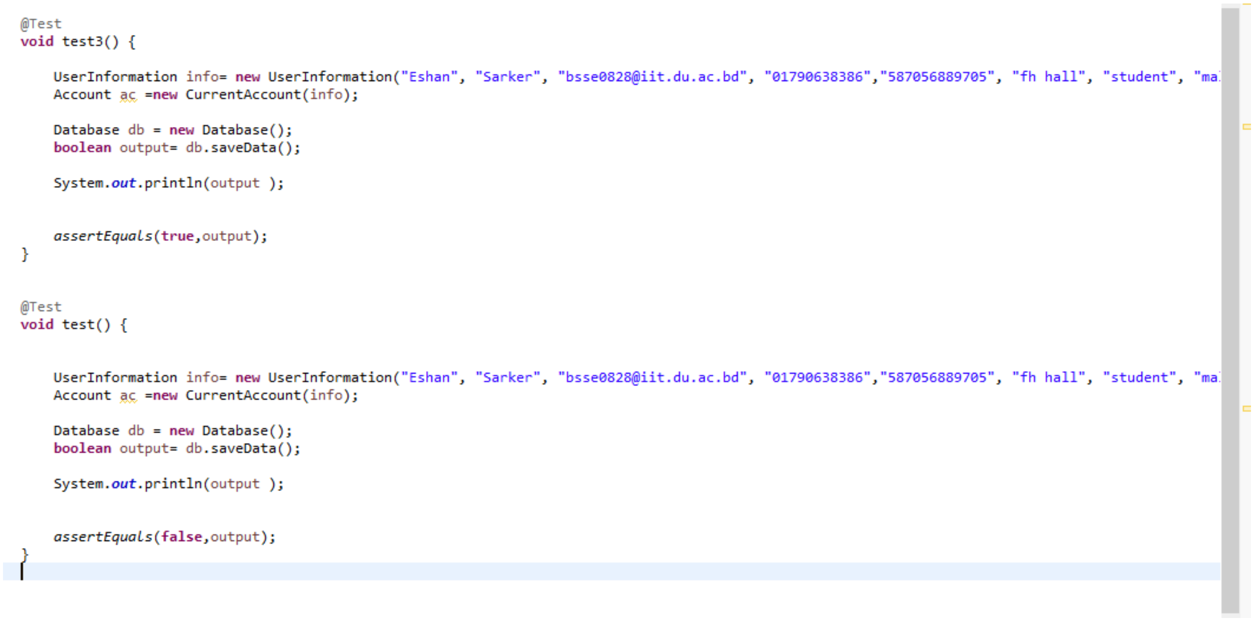
Figure 38: test case of T112,113,114,115,116,117

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T112	To check withdraw		500.0	0.0	1	failed
T113			500.0	1.0	1	passed
T114			50000.0	-49500.0	3	failed
T115	To check withdraw		50000.0	3.0	3	passed
T116			250.0	250.0	2	failed
T117			250.0	2.0	2	passed

Table 38: Test Specification of withdraw(SavingsAccount)

SaveData method from Database Class

TestCase :118,119



```
@Test
void test3() {

    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new CurrentAccount(info);

    Database db = new Database();
    boolean output= db.saveData();

    System.out.println(output );


    assertEquals(true,output);
}


@Test
void test() {


    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new CurrentAccount(info);

    Database db = new Database();
    boolean output= db.saveData();

    System.out.println(output );


    assertEquals(false,output);
}
```

Figure 39: test case of T118,119

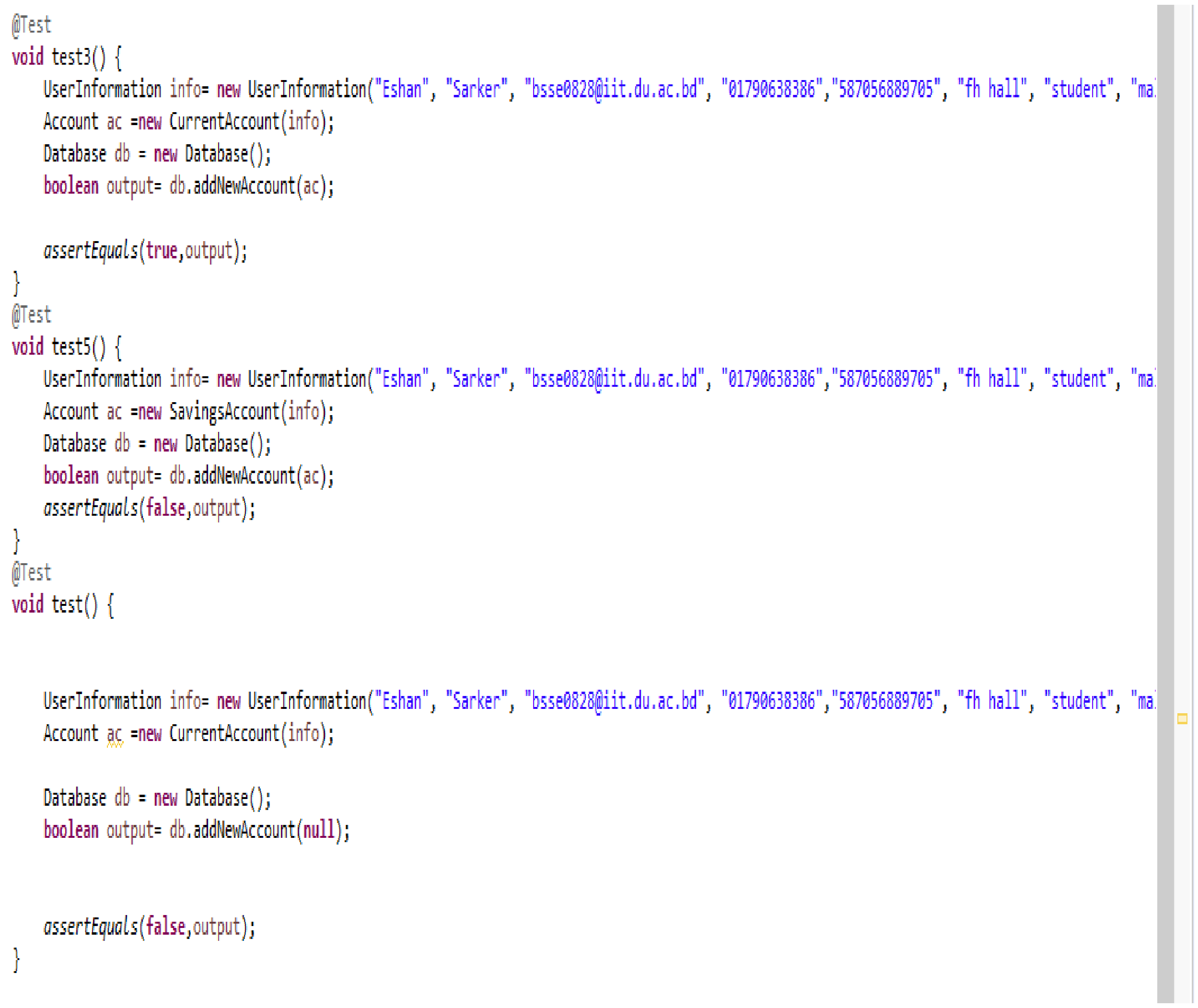
Test id	purpose	precondition	input	Expected output	Actual output	Test result
T118	To check data is save to database or not		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",nu	true	true	passed

			ll			
T119			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	false	true	failed

Table 39: Test Specification of saveData

AddAccountTesting method from Database Class

Testcase: T120,121,122



```
@Test
void test3() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new CurrentAccount(info);
    Database db = new Database();
    boolean output= db.addNewAccount(ac);

    assertEquals(true,output);
}
@Test
void test5() {
    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new SavingsAccount(info);
    Database db = new Database();
    boolean output= db.addNewAccount(ac);
    assertEquals(false,output);
}
@Test
void test() {


    UserInformation info= new UserInformation("Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "ma
    Account ac =new CurrentAccount(info);

    Database db = new Database();
    boolean output= db.addNewAccount(null);


    assertEquals(false,output);
}
```

Figure 39: test case of T120,121,122

Test id	purpose	precondition	input	Expected output	Actual output	Test result
T120	To check if the account is added to database or not		"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall",	true	true	passed

			"student", "male",null			
T121			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	false	true	failed
T121			"Eshan", "Sarker", "bsse0828@iit.du.ac.bd", "01790638386","587056889705", "fh hall", "student", "male",null	false	true	failed

Table 40: Test Specification of addAccount to Database

Chapter 4: Conclusion

As a simple project the test cases and test description produces are not sufficient.

I have tried my level best to present a readable report on Bank Management System JUnit testing ableist some shortcomings.

References
https://github.com/gbriyad/Bank-Management-System-Java last visited 11 PM 7/9/201