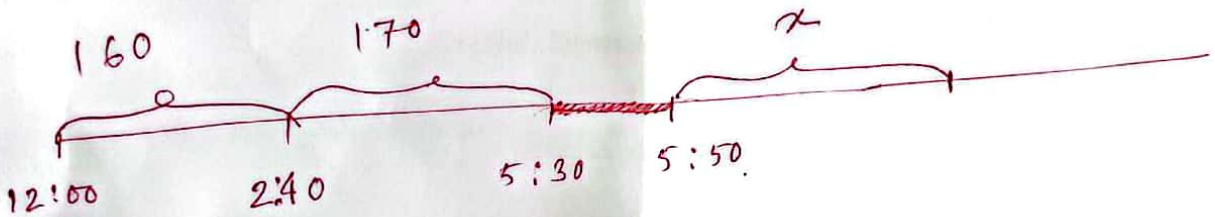
CT1 Solution

MTBF ⇒ max = 180, min = 145, Let x be the time between ~~second~~ repair and third failure

If MTBF = 145,



$$\frac{160 + 170 + x}{3} = 145$$

⇒ x = 105

∴ Likely time of crash = 7:35 AM

If MTBF = 180

$$\frac{160 + 170 + x}{3} = 180$$

x = 210

∴ Likely time of crash = 9:20 AM

∴

UNITED INTERNATIONAL UNIVERSITY (UIU)
Dept. of Computer Science & Engineering
Trimester: Summer 2022
Course No: CSE 4495 Title: Software Quality Assurance and Testing
Section: A Class Test-2

Time: 25 minutes **Marks: 20**

Name		ID	

1. Briefly describe different types of Acceptance Testing [6]
2. Consider the following function – [13] +1

   void **terminateMembership**(Set<Student> club, Student student)

   Which removes a given student from a given set of students called a **club**. For this function –

   I. Identify the parameter choices.
   II. For each choice, identify representative values.
   III. Create test specifications with expected outcomes.

Demo Solution

(2) Parameter choices for the function :-

✓ club.
✓ Student.

Representative values:-

✓ club:

i. empty set.
ii. Set with multiple students.
iii. Set with only one Student.

✓ Student:

i. existing student in the club.
ii. non existing student in the club.
iii. NULL.

Test specifications with expected outcomes

club	student	outcomes
✓ empty set	existing student	No change club will remain empty.
✓ multiple students	removed existing	valid and student removed.
✓ only one	removed one	valid and club empty.
✓ multiple students	removed nonexisting	Invalid and club not changes
✓ empty club	remove student	error and exception occur.
✓ Multiple students	remove NULL	error and exceptions.
✓ multiple students with duplicates	remove duplicates	close change duplicate and
Go on . . .		no one will be removed

UNITED INTERNATIONAL UNIVERSITY (UIU)

Dept. of Computer Science & Engineering

Trimester: Spring 2023

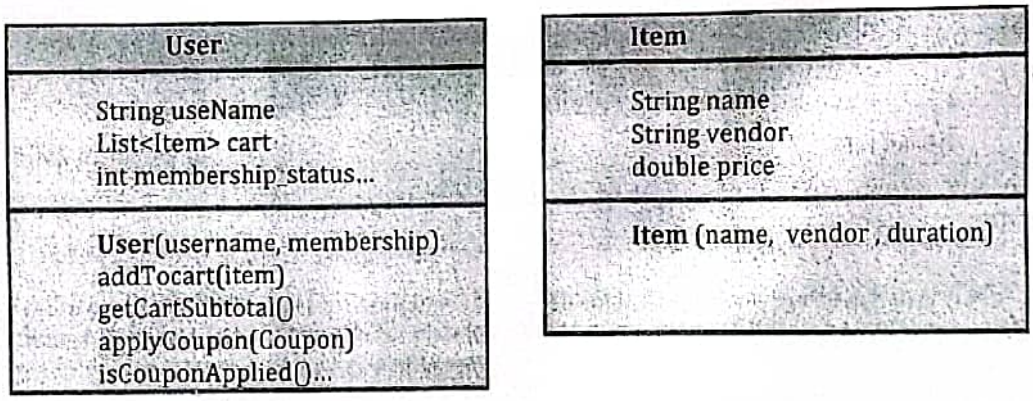
Course No: CSE 4495 Title: Software Quality Assurance and Testing

Section: A Class Test-3

Time: 25 minutes **Marks: 20**

Name		ID	

1. Consider the following class diagrams for users in an online shopping platform. Here each user has a personal cart where he/she can add items to for checkout. Each user also has a membership status. As the system is in beta phase there are only two possible values for the integer ***User. membership_status*** : 0 = 'Not membership, 1 = 'VIP membership'. The difference being users with VIP membership can apply a coupon which provides a 20% discount on the cart subtotal, whereas a non-member isn't eligible to apply. Now, devise two executable test cases for testing the **applyCoupon()** method of User class in the JUnit notation. The test case specifications are described for you - **[10+10 = 20]**



/*case-1 write a test case for a user with membership. Try to add atleast two items to the cart and apply coupon. It is expected to be applied successfully. Check the cart subtotal and coupon status for consistency. You can check whether the coupon was added successfully or not with *isCouponApplied()* method.*/

(1)

```
@Test
public void testApplyCouponWithMembership() {
    User user = new User("X", 1);

    Item item1 = new Item("Item1", "ven1", 20.0);
    Item item2 = new Item("Item2", "ven2", 30.0);

    user.addToCart(item1);
    user.addToCart(item2);

    Coupon c = new Coupon("MEMX", 20);
    user.applyCoupon(c);
```

```
double expected = item1.getPrice() + item2.getPrice();
assertEquals(expected, user.getCartSubtotal(), 0.001)
                                                  ↓ epsilon value
assertTrue(user.isCouponApplied()); }
```

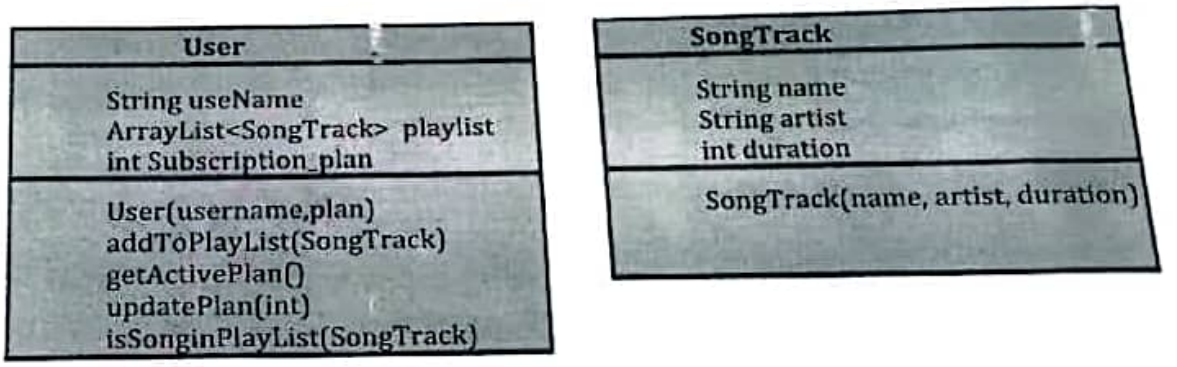
```
/*case-2 write a test case for a user with no membership Try to add atleast
two items to the cart and apply coupon. The system should throw
"NotaMemberException" with message "This coupon is not applicable for you.
Please upgrade to VIP." Also check the cart subtotal and coupon status for
consistency.*/
@Test
public void testApplyCouponWithNoMembership() {
    User user = new User("X", 0);

    Item item1 = new Item("Item1", "rem1", 20.0);
    Item item2 = new Item("Item2", "rem2", 30.0);

    user.addToCart(item1);
    user.addtocart(item2);

    Coupon c = new Coupon("memX", 20);
    Throwable exception = assertThrows(NotaMemberE-
            xception.class, () -> {user.applyCoupon(c);});

    assertEquals("This Coupon is not applicable for you.
    Please upgrade to VIP.", exception.getMessage());

    assertFalse(user.isCouponApplied());

    assertEquals(50.0, user.getCartSubtotal(), 0.001);
                                                  ↓
                                              epsilon
                                              value.
}
```

UNITED INTERNATIONAL UNIVERSITY (UIU)
Dept. of Computer Science & Engineering
Trimester: Summer 2022
Course No: CSE 4495 Title: Software Quality Assurance and Testing
Section: B Class Test-3

Time: 30 minutes **Marks: 30**

Name	Tawhidul Islam	ID	011192118

1. Consider the following class diagrams for users in a music streaming platform. Here each user has a certain subscription plan and a customizable playlist. As the system is in beta phase there are only two possible values for the integer ***User. Subscription_plan*** – 0 = 'No active plan', 1 = 'premium subscription'. The difference being users with no plan can add only upto three songs to their playlist, whereas a premium subscriber has no such restrictions.
Now, devise two executable test cases for the methods of this in the JUnit notation. The test case specifications are described for you -



/*case-1.*/ write a test case for a user with premium subscription. Try to add four songs to his/her playlist. All four songs are expected to be added to the list. You can check if the song has been added successfully with *isSonginPlayList()* method.*/

/*case-2 write a test case for a user with no subscription. Try to add four songs to his/her playlist. The first three songs are expected to be added to the list, but an exception is expected when the fourth song is added. The system should throw "PrivilegeNotGivenException" with message "You cannot add more than three songs in your current plan. Please upgrade to premium." */

CamScanner

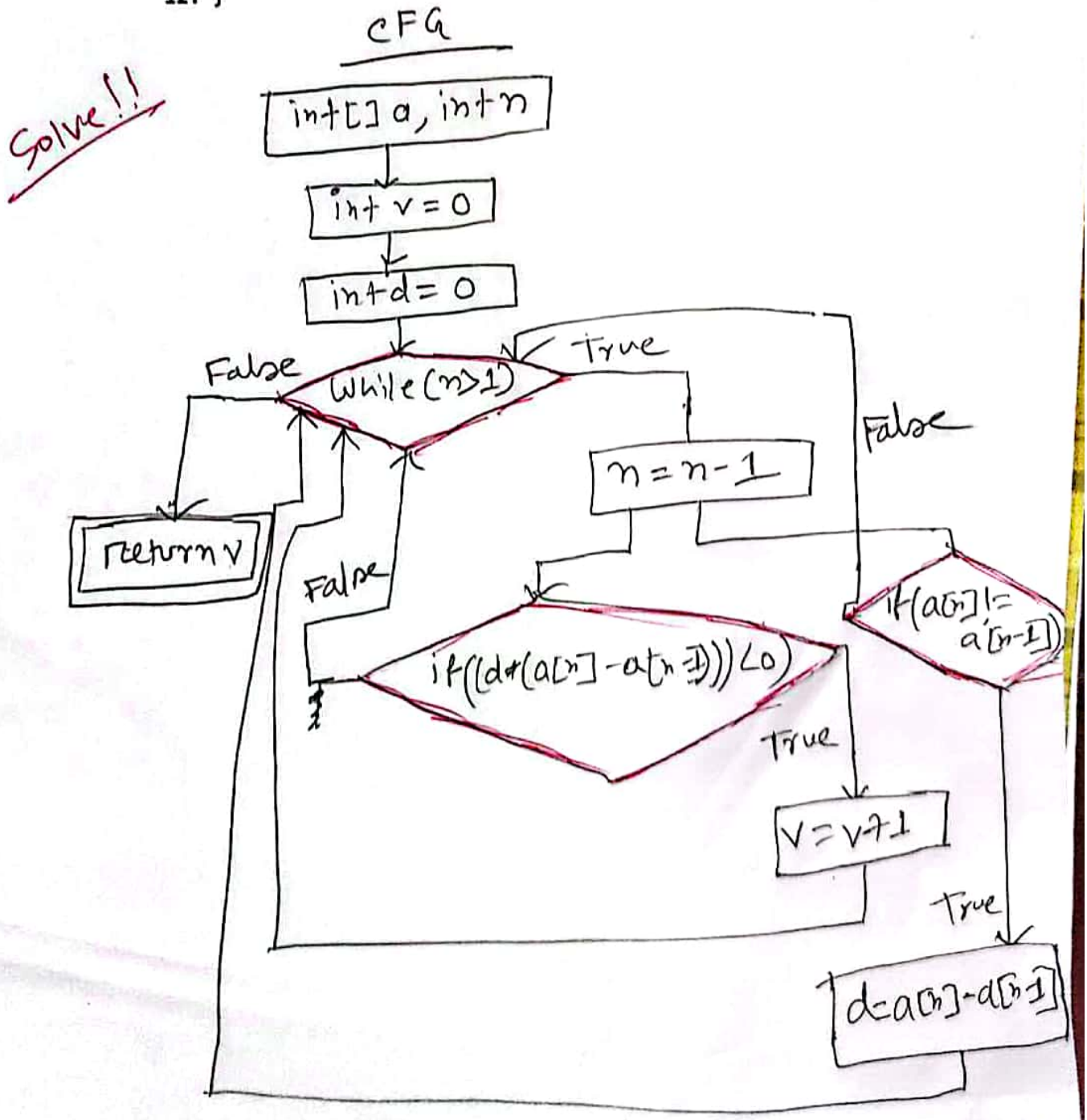
2. Draw a CFG for the following code –

```
1. public int inflections(int[] a, int n) {
2.     int v = 0; // number of inflections
3.     int d = 0; // current run direction (+/-)
4.     while (n > 1) {
5.         n = n - 1;
6.         if ((d * (a[n]-a[n-1])) < 0) // direction change
7.             v = v + 1; // => inflection point
8.         if (a[n] != a[n-1])
9.             d = a[n] - a[n-1]; // record direction
10.    }
11.    return v;
12. }
```

CamScanner

2. Draw a CFG for the following code –

```
1. public int inflections(int[] a, int n) {
2.     int v = 0; // number of inflections
3.     int d = 0; // current run direction (+/-)
4.     while (n > 1) {
5.         n = n - 1;
6.         if ((d * (a[n]-a[n-1])) < 0) // direction change
7.             v = v + 1; // => inflection point
8.         if (a[n] != a[n-1])
9.             d = a[n] - a[n-1]; // record direction
10.    }
11.    return v;
12. }
```



(a)

```
int findMax(int a, int b, int c) {
    int temp;

    if (a>b) temp=a;
    else temp=b;
    if (c>temp)
    temp = c;

    return temp;
}
```

i. Draw the control flow graph for the program ***findMax***; which finds the maximum of three integers. [3]

ii. Develop test input that will provide statement coverage. [2]

iii. Develop test input that will provide branch coverage. [2]

iv. Develop test input that will provide path coverage. [3]

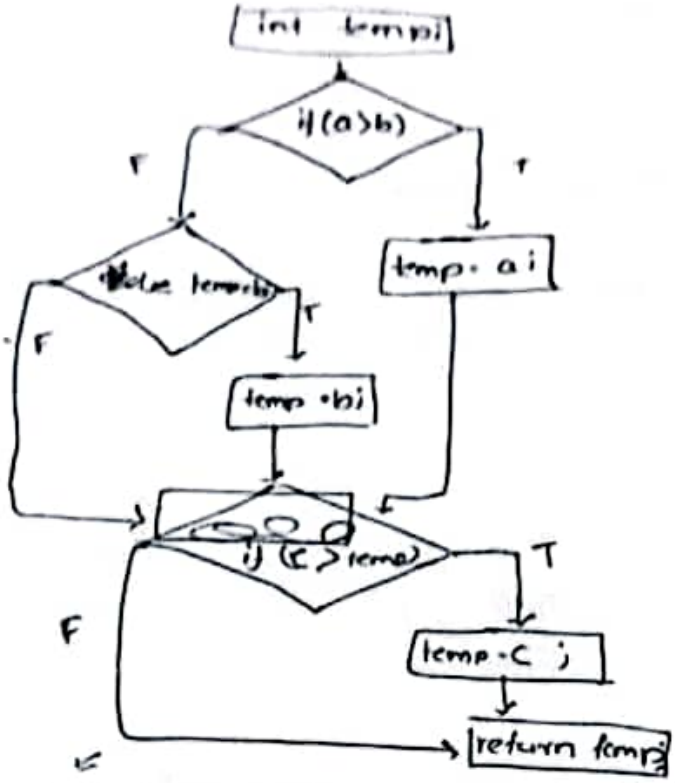
(b)

CamScanner

Date: 13th Aug 2013

Final Spring 2013

My



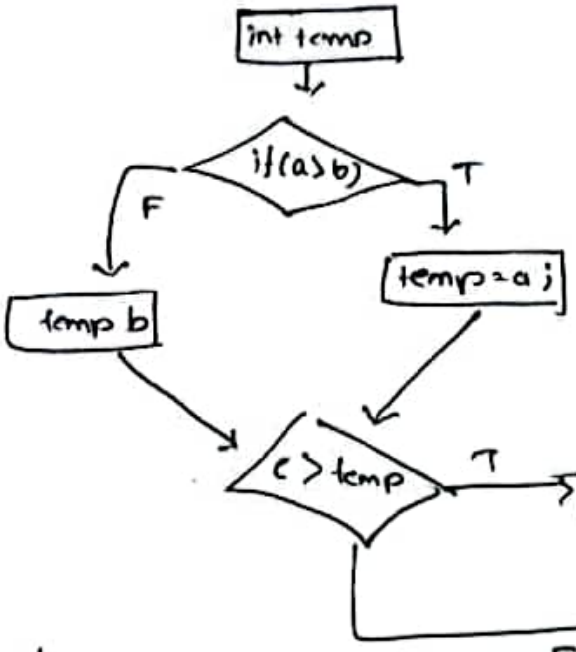
Doesn't include a statement.

All statement

(5, 2, 7)

(2, 5, 7)

Teacher



All statment

find max (5, 2, 7)

find max (2, 5, 7)

All + Branch

find_max (5, 2, 7)

find_max (2, 1, 5, 0)

Branch coverage subsumes statement coverage.

So Branch q3 answer is sufficient for statement q2 answer.

This Doesn't include a statement

Revisite of structural testing for CT-2

last final

CFG draw করা

└→ different path find out করা

  └→ branch coverage etc.

  → statement "

  → path "

i) find max এর জন্য CFG draw?

⇒