```
After this environment of the production of the 
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       For earlier recoverse higher entry for 1 their for all that creditate. These plane for following the authority of the action of the following high for all test morthwish in the first for the following the contract morthwish and the color and their higher following.
                                        on the double calculater surround (excite an article the area double same) (
Report with [ mineral [] through ()
I make two Boyald's year of troughton, the shall report ().
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       Tyri
Threed deep (10000)
                                                          cardipospuca (i) (
a prinder discrit
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         yesharali
padindadi sadidan@ent] |
yesharali ("instancian perdacileranes]");
pademali
                                                          I Martin - I Principal Control of Control of
                    AppTotal jave

    Consists once before consisting
of the had more patternatived.

                                        private rando Camidernico in aldernico;
gilitativa del
publica de la come el dej ();
su transconte má si loca entrance () ti
bost tembrano a Camidernico ();
                                                                                                                                                                                                                                                               The methods of the App., that execute 
outcomelically by underlying server to 
framework //RE is called Life cycle met 
Container Callback method
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    2 Decrees before each
and over productions
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     Marie and
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        especial after each 
sources (P. Nationalise)
                                           @Celeritors
public mid-informatic) |
within contains of Appendiculation and it
                                        Office and district consistence are red and assist to a single district to the consistence of the consistenc
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              her de une attendet
de sitte des com de décembre
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   CMARK I and Marking Villed but are instance inschool or a transposition. Nethods
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   $4 does feel and $30 in facts continue, and continued reproductly mass long blade.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               P Texture demands when a 2 methods over methods of texture.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               SMetricusti. Problem i metriculo execute only for 3 time.
Invitational and work dischinations execute for 30 minutes.
Incidigent matricular and lead to the 30 minutes.
                                                                 gives and the initial standard man $20.00 is a Congression and 1.

It is a place of the initial standard and $2.00 is a Congression and $2.00 is a Congressi
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                Oscore for - Junit Life Cycle methods
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        @Baferraid.---> To initialize Service due obj. required in all the @Ret reacheds
@Bleckill ---- To notify Service does not red used to all the @Test methods
                                                                 The read than that end and 
one complete the second or
                                                                                                                                                                                                                                                                                                                                                                                      - Sign of the second of the se
                                                                 dividualisch
public volume diverful
System und gemänfünge vertanbe herti/f
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           (Mike Sask ....e. To return Aline john om object bask to john om pool
that is used in Mike method
                                                                                        gration to
public ratio and resistancy ()
System and princip (Applications County); (
                                                                                        bankserver mit
                                                                                                                                                                                                                                                                                                                                                                                                        of the second
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           :> The lava class that contains the bilogic is called Sendre class
:> The lava class that contains the perdistance is called BAD class
:> We do not write test cases (ØTest methods) separately for DAD class
methods becer DAD class methods are levabled from Service class method
to Tasting write methods indirectly best DAD class methods have
                                                            dependential in parts and the
                                                               How to test private method of the service diss?

***Dil ye can not comprise methods of service dissiplicity, do call
there from public methods of the revision dissiplicity and write just the down
on the public method; to perform the unit testing.
                                                  Firamples on assertTrac() and assertFalse() methods
                                                                                        In Service days
                                                                      public boolean isPallendronesString stri (
                                                                                                          P(corneal | | colough() = 0 | | conquisignoreduce| ")|
three and Equilibration biographic [low billions"];
                                                                                                                    Sing cone Strange String Salite (str) reverse() in String (;
P(maquality contain) reventor()
return true;
                                                                                                                 else
returnfalses
                                                                                 @Text
public void texts@dendroneWithWikitopats@ [
coord True(service is PallendroneC mailors %);
                                                                                                          The public void treds Pollendrone With bindron (0,0) = (\cos i \theta \sin i \theta \sin \theta \cos \theta)^{-1} [1] assert Theorem's Pollendrone (0,0) = (\cos i \theta \sin i \theta \sin \theta \cos \theta)^{-1} [1]
                      Assignment: Write Junit test cases for calculateCompoundIntrestAmount(......) method
                                                               More example on essertXxx() methods
                                   //Service class
  package com.nt.service
import java.util.Date;
import com.nt.model.Students
public class MoreOperationsService (
                      String reveseStr-new StringBuilder(str).reverse().toString();
if(reveseStr-equalsignoreCase(str))
                         // 8.method2
public Date showDateByMonth(int month) (
iffmonths=1 && months=12|
return new Date();
```

Passeple App. using Test constitution based life systematicals

```
Example App using Test annotation based Life cycle methods
//BankService.java
package com.nt.service;
public class BankService {
public double calcintrestAmount(double pamt,double rate,double time) {
if(pamt<=0 | | rate<=0 || time<=0)
try {
}
throw new IllegalArgumentException("invalid inputs");
Thread.sleep(30000);
catch(Exception e) {
e.printStackTrace();
//calculate the Simple Intrest amount
return pamt*rate*time/100.0f;
To write common logic only for 1 time for all test methods.. then place in @BeforeAll method.. similarly place
cleanup logic for all test methods in @AfterAll method.. These methods must be taken as static methods
AppTest.java
public class AppTest
{
private static BankService bankService; @BeforeAll
public static void setUp() {
System.out.println("AppTest.setUp()");
bankService=new BankService();
The methods of the App, that executes automatically by underlying server /container/ framework/JRE is
called Life cycle method or Container Callback method
@BeforeEach
public void beforeTest() {
System.out.println("AppTest.beforeTest()");
}
@Test
public void testCalcintrestAmountWithBigValues() {
double expected=287999.0;
```

```
double actual-bankService.calcintrestAmount (1200000.0, 2.0, 12.0); assertEquals(expected, actual, 1.0f);
}
@Test
@BeforeAll
public static void setUpOnce() {
System.out.println("TestBankLoanService.setUpOnce()");
service=new BankLoanService();
}
@AfterAll
public static void clearOnce() {
System.out.println("TestBankLoanService.clearOnce()");
service=null;
Delta value ---> The acceptable difference value
between actual result and the expected result
(very useful while scientific busines methods)
public void testCalcIntrestAmountWithSmallValues() {
double expected=1200.0;
double actual bankService.calcintrestAmount(10000.0, 1.0, 12.0); assertEquals(expected, actual);
}
JUNIT 5-LifeCycle
Call-back Annotation
@BeforeAll
Testcase 1
@Test --- Testcase 2
Executes once before executing
all the test cases (@Test methods)
Executes before each
Test case
(@Test method)
executes after each
test case (@ Test method)
```

```
Executes once at the end of
the all the Test cases (for all @Test methods)
@AfterAll and @BeforeAll methods are TestCase class level one tine executing methods test @BeforeEach
and @AfterEach methods are each case level repeatedly executing blocks
as
if Testcase class is having 20 methods test methods
then
=>@BeforeAll, @AfterAll methods execute only for 1 time =>@BeForeEach, @AfterEach methods execute for
20 times on 1 per each test method basis
@Test
public void testCalcIntrestAmountWithZeroOrNegetiveValues() {
The expected excption });
assertThrows(IllegalArgumentException.class, ()->{
bankService.calcintrestAmount (120000,-1,-20);
Usecase for
Junit Life Cycle methods
}
@Test
public void testCalcintrestAmount Timer() {
assertTimeout(Duration.ofMillis(30015), ()->{bankService.calcIntrestAmount (100000.0,2.0, 12);});
}
The max time that method
@AfterEach
can complete the execution
public void afterTest() {
System.out.println("AppTest.afterTest()");
@AfterAll
public static void tearDown() {
System.out.println("AppTest.tearDown()");
bankService=null;
com.nt.service
JunitProj02
```

@AfterAll

```
> BankService.java
#src/test/java
com.nt.tests
> AppTest.java
JRE System Library [JavaSE-17]
Maven Dependencies
>junit-jupiter-api-5.10.0.jar - C:\Users\NATAR >opentest4j-1.3.0.jar - C:\Users\NATARAJ\.m?
> junit-platform-commons-1.10.0.jar - C:\Use >apiguardian-api-1.1.2.jar - C:\Users\NATARA
>junit-jupiter-params-5.10.0.jar - C:\Users\N/
src
> target
@BeforeAll ----> To initialize Service clss obj required in all the @Test methods @AfterAll ----> To nullify
Service class obj ref used in all the @Test methods
@BeforeEach Tgetach new jdbc con object from the jdbc con pool required in @Test method execution
@AfterEach----> To return/close jdbc con object back to jdbc con pool that is used in @Test method
pom.xml
}
dependencies in pom.xml file
<!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api --> <dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter-api</artifactId>
<version>5.10.0</version>
=> The Java class that contains the b.logic is called Service class
=> The Java class that contains the persistence is called DAO class
=> We do not write test cases (@Test methods) separately for DAO class
methods becoz DAO class methods are invoked from Service class methods So Testing service methods
indirectly test DAO class methods
<scope>test</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-params -->
<dependency>
<groupId>org.junit.jupiter</groupId>
<artifactId>junit-jupiter-params</artifactId>
```

src/main/java

<version>5.10.0</version>

```
</dependency>
How to test private method of the service class?
ans) we can not test private methods of service calss directly.. So call
them from public methods of the service class and write junit test cases on the public methods to perform
the unit testing
Examples on assertTrue() and assertFalse() methods
======
=======
========
In Service class
public boolean isPallendrome(String str) {
if(str==null || str.length()==0 || str.equalsIgnoreCase("""")) throw new IllegalArgumentException("Invalid
Input");
String reveseStr=new StringBuilder(str).reverse().toString(); if(str.equalsIgnoreCase(reveseStr))
}
return true;
else
return false;
In Testcase class
@Test
public void testlsPallendromeWithValidInputs() { assertTrue(service.isPallendrome("madam"));
}
@Test
public void testls PallendromeWithInValidInputs() {
assertFalse(service.isPallendrome("madam1"));
}
@Test
public void testlsPallendromeWithNoInputs() {
assertThrows (IllegalArgumentException.class,()->{ service.isPallendrome(""); });
Assignment :: Write Junit test cases for calculateCompoundIntrestAmount(-,-,-) method
More example on assertXxx() methods
//Service class
_____
package com.nt.service;
```

<scope>test</scope>

```
import java.util.Date;
import com.nt.model.Student;
public class More OperationsService {
//=B.method1
public boolean isPallendrome(String str) {
if(str==null || str.length()==0 || str.equalsIgnoreCase(""")) throw new IllegalArgumentException("Invalid String
"); //reverse the string
String reveseStr=new StringBuilder(str).reverse().toString(); if(reveseStr.equalsIgnoreCase(str))
return true;
else
return false;
// B.method2
public Date showDateByMonth(int month) {
if(month>=1 && month<=12)
else
return new Date();
return null;
// Test case Class
//MoreOpertionsService Test
package com.nt.service;
import static org.junit.jupiter.api.Assertions.assertFalse; import static
org. junit. jupiter. api. Assertions. assert Not Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert Null; import \ static \ org. junit. jupiter. api. Assertions. assert \ org. junit. assert \ org. jun
static org.junit.jupiter.api.Assertions.assertTrue; import static org.junit.jupiter.api.Assertions.assertSame;
import static org.junit.jupiter.api.Assertions.assertNotSame;
import java.time.LocalTime;
import java.util.Date;
import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
public class More OperationsService Test {
private static MoreOperationsService service;
@BeforeAll
public static void setUpOnce() {
System.out.println("MoreOperationsServiceTest.setUpOnce()");
```

```
service=new More OperationsService();
}
}
@Test
public void testis PallendromeWithValidData() {
String msg="madam";
assertTrue(service.isPallendrome(msg));
}
@Test
public void testis PallendromeWithInValidData() {
String msg="nit";
assertFalse(service.isPallendrome(msg));
}
@Test
public void testShowDate ByValidMonth() {
Date actual-service.showDateByMonth(1);
assertNotNull(actual);
}
@Test
public void testShowDateByInValidMonth() {
Date actual-service.showDateByMonth(-1);
assertNull(actual);
@Test
public void testSingleton Runtime() {
Runtime time1=Runtime.getRuntime();
Runtime time2=Runtime.getRuntime();
assertSame(time1,time2);
}
@Test
public void testSingletonLocalTime() {
LocalTime time1=LocalTime.now();
LocalTime time2=LocalTime.of(10, 20);
assertNotSame(time1,time2);
@AfterAll
public static void tearDownOnce() {
```

```
System.out.println("MoreOperationsServiceTest.tearDownOnce()");
service=null;
}
```