**IONJavaTest**

**Thistestcontains9questionsandonecodingexercise.**

**You have1.5hourtocompletethetest,withaguidelinefortesttotake20minsand70minutesforcoding exercise.YouareexpectedtohaveJavadevelopmentenvironmentreadybeforestartingtoworkonthe exercise.**

**Youareresponsiblefortimemanagementyourselfandifyousubmitaftertimehasexpiredyourscorewillbe adjusted.**

**Donotclosebrowserwindowuntilyouhavesubmittedtestresults.Ifyouhavedonesobyaccidentpleaseredo theformbutbeawarethattimewillbecalculatedfromthefirsttimeyouhaveopened theform.**

**Exercisecodeshouldbesubmittedseparatelyaszipfilewithinstructionsonhowtobuildit.**

**Q1 Consider the following code snippet**

public class Test1 {

public static void main(String[] args){

String str1 = "One";

String str2 = "One";

String str3 = new String("One");

Test1 test = new Test1();

boolean isEqual = test.compare(str1, str2);

test.display(isEqual);

isEqual = test.compare(str2, str3);

test.display(isEqual);

}

public boolean compare(String str1, String str2){

return (str1 == str2) ? true : false;

}

public void display(boolean isEqual){

System.out.println("isEqual =" + isEqual);

}

}

**What would be the output of the above:**

**(CheckONEthatapplies)**

**(a) true true  
(b) true false  
(c) false false  
(d) false true**

**Q2 Consider the following code snippet**

class Level1Exception extends Exception {}

class Level2Exception extends Level1Exception {}

class Level3Exception extends Level2Exception {}

class Purple {

public static void main(String args[]) {

try {

throw new Level1Exception();

}catch (Level1Exception e)

{

System.out.println("Level1");

}catch (Level2Exception e) {

System.out.println("Level2");

}catch (Level3Exception e) {

System.out.println("Level3");

}

}

}

**What is the output of the code above:**

**(CircleONEchoice)**

1. **Prints Level1**
2. **Prints Level2**
3. **Prints Level3**

**(d) Compile-time error**

**Q3 The GenericFruit class declares the following method.**

public void setCalorieContent( float f )  
  
**You are writing a class Apple to extend GenericFruit and wish to add methods which overload the method in GenericFruit.  
Which would constitute legal declarations of overloading methods?**

**(CheckANYthatapply)**

**(a) protected float setCalorieContent(String s )  
(b) protected void setCalorieContent( float x )(c)  
(c) public void setCalorieContent( double d )  
(d) public void setCalorieContent(String s ) throws NumberFormatException**

**Q4 Consider the following class DerivedDemo**

public class DerivedDemo extends Demo{

int M, N, L ;

public DerivedDemo( int x, int y ){

M = x ; N = y ;

}

public DerivedDemo( int x ){

super( x );

}

}

**Which of the following constructor signatures MUST exist in the Demo class for DerivedDemo to compile correctly?**

**(CheckANYthatapply)**

**(a) public Demo( int a, int b )  
(b) public Demo( int c )  
(c) public Demo( )**

**Q5 Write an implementation for an immutable class employee with attributes name (String), age (int) and date\_of\_joining (Date).**

**Ans:**

**final public class Employee**

**{**

**final private String name;**

**final private int age ;**

**final private Date date\_of\_joining;**

**}**

**Q6 Consider the following code snippet**

class TSafe1 {  
  
 private int a;  
  
 public synchronized void increment(){  
 a++;  
 }  
  
 public int getVal(){  
 return a;  
 }  
}

**Which of the following statements are true?**

**(CheckANYthatapply)**

**a. The class works correctly in a single threaded environment.  
b. The class will work with multi thread access if no methods are called concurrently.  
c. Making getVal() method synchronized will make it thread safe.  
d. The class is thread safe.**

**Q7 Which of the following statements are true?**

**(CheckANYthatapply)**

**a. Iterating over a hashmap can throw ConcurrentModificationException.  
b. Iterating over concurrentHashMap can throw ConcurrentModificationException.  
c. Iterating over concurrentHashMap will never throw ConcurrentModificationException but the iteration may not reflect latest state.  
d. Iterating over a LinkedList can throw ConcurrentModificationException.**

**Q8 Which of the following is a valid case for single instance CountdownLatch?**

**(CheckANYthatapply)**

**a. Waiting for 2 jobs running in different threads to get completed.  
b. Waiting for 2 jobs running in different threads to get completed and then waiting for another 2 jobs likewise.  
c. Implementing a thread pool.  
d. Executing tasks in parallel.**

**Q 9 Which of the following is true about "synchronized" keyword when used with a method or block?**

**(CheckANYthatapply)**

**a. The lock acquired when entering a "synchronized" method is reentrant.  
b. "synchronized" block can be used with a thread instance. i.e. (synchronized(new Thread()))  
c. The lock acquired when entering a "synchronized" method is fair.  
d. Its possible that both static synchronized and non-static synchronized method can run simultaneously or concurrently because they lock on different object**

**Coding Exercise**

**We need to create a simple bank account management system with details as given below. There is no GUI so it is a simple java application which provides the output as described below and no input required from user.**

1. **A Bank Account has the following attributes:-**

**- Id**

**- Name**

**- Balance**

1. **Actions/Operations allowed on the Bank Account are given below:**

**- Create a bank account**

**- Delete a bank account**

**- Withdraw Money from a bank account**

**- Deposit Money in the bank account**

**- Get Current Balance for a bank account**

**- Transfer money from one account to another.**

1. **Basic Test - Guidelines**

**- Handling for negative cases for example**

**- Negative balance is not allowed**

**- Cannot delete a bank account which has non-zero balance**

**- Cannot withdraw/transfer more than existing balance**

**- The code should compile and give correct result for the following Test.**

**- Create new account for User “Mohan” and get balance for Account (should be 0).**

**- Deposit 10000 Rs in the account of Mohan and Get current balance (should be 10000)**

**- Withdraw 5000 Rs from the account of Mohan and Get current balance (should be 5000)**

**- Try to Withdraw 6000 Rs from the account of Mohan (should give Error) and then get the current balance (which should still be 5000)**

**- Try Deleting the account for Mohan which should give an error.**

**- Withdraw 3000 Rs from the account and get the current balance (which should be 2000)**

**- Create another account for Mohan**

**- Transfer 3000 from Account1 to Account2 (should give Error)**

**- Transfer 2000 from Account1 to Account2 and get the balance (which should be 0)**

**- Delete the account 1 for Mohan**

1. **Additional Features / Advanced operations**

**- Get Last Ten Transactions for Account**

**- Get Transactions by Date for Account**

**- Get Transactions by Amount for Account**

1. **Guidelines for Advanced Features**

**- Implement the multi-threading solution which handles concurrent operations on same or different account and atomicity on operations like transfers which involve multiple accounts.**

**- Add feature to get consolidated statement (at least total balance across accounts) for all accounts for a person**

**- Test the multi-threaded version on the following lines:**

**- Create 2 accounts for User Ram and perform the following operations.**

* **Deposit 5000 Rsin 2 different accounts and get Current Balance for each account such thatprocessing the request for each account is done concurrently.**
* **Withdraw 2000 Rs from 2 different accounts and get Current Balance for each account such thatprocessing for each account is done concurrently.**
* **In one thread perform transfer so that 2000 Rs is transferred from Account 1 to 2 and show balance for both accounts while in a separate thread you withdraw 2000Rs from Account 2and then show its balance**
* **Get consolidated statement for both accounts for the user.**