JEE FS Mod 1 MTT Set 2

Section:

* 1
* 2

**Time Left :** **65 : 08**

**List View**



*Active     Attempted     Not Attempted*

**Q1**

Why should a program perform unit testing?

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

Unit testing is used to check weather the whole application functions as per the requirements after it is completely developed

Unit testing provides immediate feedback on code

Unit testing helps developers find errors in code

Unit testing is used to test the non functional requirements of the software

Output:

x

Developer Tools

12

x

Core Java

345678910111213141516171819202122232425262728293031323334353637383940

**Q1**

Why should a program perform unit testing?

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

Unit testing is used to check weather the whole application functions as per the requirements after it is completely developed

Unit testing provides immediate feedback on code

Unit testing helps developers find errors in code

Unit testing is used to test the non functional requirements of the software

Output:

**Q2**

How will you test a piece of code to check that an exception is thrown correctly using junit 4 annotations?

*Single Choice - Select one correct answer from the options list.*

Use @Exception annotation and specify the type of exception that should be thrown for the method to test for expected exceptions

Use expected parameter of @Exception annotation within the try -catch block within the test method to test the exception that should be thrown

Use @Exception annotation in the test method to test the exception that shoule be thrown

Use expected parameter in @Test annotation of the test method to test the exception that should be thrown

Output:

**Q3**

1. import java.util.\*;  
2. class Demo{  
3. public static void main(String args[])  
    {  
4.  ArrayList <String> list=new ArrayList<String>();  
5. list.add("One");  
6. list.add("Two");  
7. System.out.println(list.contains(new String("One"));  
8.  System.out.println(list.indexOf("Two"));  
9.  list.clear();  
10.  System.out.println(list);  
11. System.out.println(list.get(1));  
12.  }  
13.  }  
  
What is the output of below code ?

*Single Choice - Select one correct answer from the options list.*

Line 7 prints true.  
Line 8 prints -1.  
Line 9 removes all elements of the list .  
Line 10 prints null.  
Line 11 throws Exception

Line 7 prints false  
Line 8 prints 1.  
Line 9 will assign null to list  
Line 10 prints null.  
Line 11 throws Exception

Line 7 prints false  
Line 8 prints 1  
Line 9 removes all elements of the list .  
Line 10 prints [ ]  
Line 11 thorws Exception

Line 7 prints true.  
Line 8 prints -1.  
Line 9 removes all elements of the list .  
Line 10 prints [ ]  
Line 11 displays null

Output:

**Q4**

Choose the correct code to create an integer stream from a collection.

*Single Choice - Select one correct answer from the options list.*

int numbers[]={2,3,4,5};  
Stream<Integer> stream4=IntStream.of(numbers);  
  
stream4.forEach(System.out::println);

IntStream numbers=IntStream.of(2,3,4,5);

IntStream range = IntStream.range(1, 6);

int numbers[]={2,3,4,5};  
IntStream numbers=IntStream.of(numbers);

Output:

**Q5**

Which of the following are valid lines of the code to define a multidimentional array of int?

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

int[][] array1= {{ 1,2,3}, {},{1,2,3,4}};

int [] [] array2= new array ()   
{{ 1,2,3}, {},{1,2,3,4}};

int [] [] array3 = {1,2,3},{},{1,2,3,4} ;

int [] [ ] array4= new int [2] [];

Output:

**Q6**

When a class extends the Thread class ,it should override \_\_\_\_\_\_\_\_\_ method of Thread class   
to start the execution of thread.

*Single Choice - Select one correct answer from the options list.*

start()

run()

init()

Thread()

Output:

**Q7**

class X{  
   X()  
{  }  
   private void one()  
{      }  
}  
  
public class Y extends X {  
   Y()  
{  }  
   private void two()   
{ one();}  
  
   public static void main(String args[])  
{  
       new Y().two();  
   }  
}  
  
What changes will make this code compile ?

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

Adding the public modifier to the declaration of class x

Adding the protected modifier to the x() constructor

Changing the private modifier on the declaration of the one() method to protected

Removing the private modifier from the two () method

Output:

**Q8**

Which of the below statements are true for Class in Java?

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

A source file can have only one public class.

A file can have more than one public class

File with public class have no naming restriction

package and import statements apply to all classes in the file

Output:

**Q9**

Given:  
  
5.// insert the code here   
6.public abstract void bark();  
7.}  
  
8.//insert the code here   
9. public void bark()  
      {  
10. System.out.println("woof ");  
11. }  
12. }  
  
What code should be inserted at line no 5 and 8 to code to compile?

*Single Choice - Select one correct answer from the options list.*

5. class Dog {  
9.public class Poodle extends Dog{

5. abstract Dog {  
9.public class Poodle extends Dog{

5. abstract class Dog {  
9.public class Poodle extends Dog{

5. interface Dog {  
9.public class Poodle extends Dog{

Output:

**Q10**

Return type of "λE" is\_\_\_\_\_\_\_\_\_\_\_\_\_  !

*Single Choice - Select one correct answer from the options list.*

Function method

Functional Interface

abstract methods

static method

Output:

**Q11**

import java.io.\*;  
4. class Vehicle { }  
5. class Wheels { }  
6. class Car extends Vehicle implements Serializable { }  
7. class Ford extends Car { }  
8. class Dodge extends Car {  
9. Wheels w = new Wheels();  
10. }  
  
Instances of which class(es) can be serialized? (Choose all that apply.)  
  
A. Car  
B. Ford  
C. Dodge  
D. Wheels  
E. Vehicle

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

Car

Ford

Vehicle

Dodge

Output:

**Q12**

What is the output of below code ?  
  
class  JavaGuru{  
   public static void main(String args[])  
{  
      String str="game";  
       str.replace('a','Z').trim().concat("Aa");  
       str.substring(0,2);  
       System.out.println(str);  
  }  
}

*Single Choice - Select one correct answer from the options list.*

gZmeAa

game

gZm

gZ

Output:

**Q13**

Which of the statements are correct ?

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

An ArrayList offers a resizable array which is easily managed using the methods it provides .

An ArrayList requires you to specify the total elements before you can store any elements in it.

An ArrayList can store objects of any type

ArrayList is synchronized.

Output:

**Q14**

Which of the following code snippet can be used to create a dir by the name dir3 and   
create a file named file3 inside dir3 ?

*Single Choice - Select one correct answer from the options list.*

import java.io.File;  
class Maker {  
public static void main(String[] args)   
{  
try {  
File dir = new File("dir3");  
dir.newDir();  
File file = new File(dir, "file3");  
file.createFile();  
} catch (Exception x)   
{ }  
} }

import java.io.File;  
class Maker {  
public static void main(String[] args)   
{  
try {  
File dir = new Files("dir3");  
dir.makeDirectory();  
File file = new Files("file3");  
file.createNewFile();  
} catch (Exception x)   
{ }  
} }

import java.io.File;  
class Maker {  
public static void main(String[] args)   
{  
try {  
File dir = new File("dir3");  
dir.mkdir();  
File file = new File(dir, "file3");  
file.createNewFile();  
} catch (Exception x)   
{ }  
} }

import java.io.File;  
class Maker {  
public static void main(String[] args)   
{  
try {  
File dir = new Files("dir3");  
dir.createDir();  
dir.createFile("file3");  
} catch (Exception x)   
{ }  
} }

Output:

**Q15**

Which two statements correctly describe checked exception?

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

These are exceptional conditions that a well-written application should anticipate and   
recover from.

These are exceptional conditions that are external to the application, and that the application   
usually cannot anticipate or recover from

These are exceptional conditions that are internal to the application, and that the application   
usually cannot anticipate or recover from.

Every class that is a subclass of RuntimeException and Error is categorized as checked exception.

Every class that is a subclass of Exception, excluding RuntimeException and its subclasses,   
is categorized as checked exception.

Output:

**Q16**

List<String> list = new ArrayList<>();  //Line 1  
list.add("Hello");  
list.add("World");  
Consumer<List<String>> consumer = Collections ::sort; //Line 2  
consumer.accept(list);      
System.out.println(list);  
  
Refer to the above code snippet; what will be the output in console?

*Single Choice - Select one correct answer from the options list.*

[Hello, Java, Spring, World]

Displays the list as it is since sort method internally creates new List and stores sorted elements in it

Compilation error at Line number 1, as generics not specified

Compilation error at Line number 2, as sort method return type is void

Output:

**Q17**

Which is the base class of both checked and unchecked exceptions?

*Single Choice - Select one correct answer from the options list.*

java.lang.Error

java.lang.Exception

java.lang.RuntimeException

java.lang.Throwable

java.lang.System

Output:

**Q18**

Given:  
  
try { int x = Integer.parseInt("two"); }  
  
Which could be used to create an appropriate catch block? (Choose all that apply.)

*Single Choice - Select one correct answer from the options list.*

ClassCastException

IllegalStateException

NumberFormatException

IllegalArgumentException

Output:

**Q19**

Given:  
  
public static void printNumbers()   
{  
Set set = new TreeSet();  
set.add("2");  
set.add(3);  
set.add("1");  
Iterator it = set.iterator();  
while (it.hasNext())  
System.out.print(it.next() + " ");  
}  
  
Which Statement is true ?

*Single Choice - Select one correct answer from the options list.*

the method will print 1 2 3

The method will print 1 2

Compilation error as type argument not specified

The method will throw runtime exception.

Output:

**Q20**

Consider the following ArrayList of string.  
  
List<String> words = Arrays.asList("location Pune");  
  
Which of the following is the correct way of finding the number of  characters in each word   
present in the ArrayList?

*Single Choice - Select one correct answer from the options list.*

words.streams().map(str->str.nextElement()).forEach(System.out :: println);

words.stream().map(str->str.size()).forEach(System.out :: println());

words.stream().map(str->str.size()).forEach(System.out :: println);

words.stream().map(str->str.length()).forEach(System.out :: println);

Output:

**Q21**

What is the output for the below code ?  
  
class A implements Runnable{  
      public void run()  
{  
            System.out.println(Thread.currentThread().getName());  
      }  
}  
  
1.       public class Test{    
2.       public static void main(String... args)  
          {   
3.             A a = new A();  
4.        Thread t = new Thread(a);  
5.        t.setName("good");  
6.        t.start();  
7.       }  
8. }

*Single Choice - Select one correct answer from the options list.*

good

null

Compilation fails with an error at line 5

Compilation succeed but Runtime Exception

Output:

**Q22**

Which declarations will compile without error?  
  
(a) Map<Integer, HashMap<Integer, String>> map2  
= new HashMap<Integer, HashMap<Integer, String>>();  
(b) Map<Integer, Integer> map3 = new HashMap<Integer, Integer>();  
(c) Map<? super Integer, ? super Integer> map4  
= new HashMap<? super Integer, ? super Integer>();

*Single Choice - Select one correct answer from the options list.*

all will compile without error

a and b

a and c

b and c

Output:

**Q23**

class Hello{  
    public static void main(String []args)  
{  
      System.out.println("Hello .."+args[1]);  
   }  
}  
public class World {   
    public static void main(String args[])  
{  
       Hello.main(args);  
    }  
}  
  
And the commands :  
Javac World.java  
java World Beginner Dude  
  
What will be the result?

*Single Choice - Select one correct answer from the options list.*

Hello.. Beginner

Hello.. Dude

Exception will be thrown

The program thorws runtime erorr

Output:

**Q24**

Consider the following code and predict the output:  
  
public class NoSuchEmployeeException extends Exception {  
 public NoSuchEmployeeException(String msg)  
{    
  // line 1   
System.out.println("In constructor ");  
//line 3 }  
}  
  
public class Test{  
public static void testEmployee(int empno) throws NoSuchEmployeeException{  
if(empno==0)throw new NoSuchEmployeeException("emp id not found");  
}  
 public static void main(String[] args)   
{  
try {  
    testEmployee(0);  
    } catch (NoSuchEmployeeException excep)   
    {  
       System.out.println(excep.getMessage());  
    }  
    }  
}

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

In constructor  
null

NullPointerException

if super(msg) is inserted at line 1; the following two lines get printed   
In constructor   
emp id not found

if super(msg) is inserted at line 3; the following two lines get printed   
In constructor   
emp id not found

Output:

**Q25**

What is the output of following code ?  
  
public class JavaString{  
public static void main(String args[])  
{  
String str1= new String("Java");  
String str2= new String ("Java");  
String str3= "Java";  
String str4= "Java";  
do{  
System.out.println(str1.equals(str2)) ;  
}  
while(str3 == str4 );  
}

*Single Choice - Select one correct answer from the options list.*

true printed once

false printed once

true printed in an infinite loop

false printed in an infinite loop

Output:

**Q26**

Which of the following best describes the data written by an ObjectOutputStream?

*Single Choice - Select one correct answer from the options list.*

Bytes and other Java primitive types.

Object hierarchies

Object hierarchies and Java primitive types.

Single objects

Single objects and Java primitive types.

Output:

**Q27**

1.public class App {  
2.// Insert code here  
3.System.out.print("Welcome to the world of 4.Java");  
5.}  
6.}  
  
Which code fragments, when inserted independently at line 2 // Insert code here,   
enable the program to execute and print the welcome message on the screen?

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

static public void main (String [] args)   
{

static void main (String [] args)   
{

public static void Main (String [] args)   
{

public static void main (String... args)   
{

public void main (String [] args)   
{

Output:

**Q28**

Identify the error if any in the following program?  
  
Assume that All the classes and interface are in same package.  
  
interface Vegetarian  
{  
}  
abstract class Animal  
{  
  protected void live()  
 {  
  System.out.println("Live");  
 }   
}  
class Deer extends Animal implements Vegetarian  
{       }  
  
public class MainClass  
{   
 public static void main(String[] args)   
 {             
  new Deer().live();    
 }  
}

*Single Choice - Select one correct answer from the options list.*

There is an error in the program as interface Vegetarian should contain atleast one abstract method

There is an error in the program as the abstract class Animal must contain atleast one abstract method

There is an error in the program as method live() is declared protected which is not allowed in   
an abstract class

The program shows error as protected void live() is not overridden by the class Deer

The program shows output : Live

Output:

**Q29**

Given:  
  
3. class SubException extends Exception { }  
4. class SubSubException extends SubException { }  
5.  
6. public class CC { void doStuff() throws SubException { } }  
7.  
8. class CC2 extends CC { void doStuff() throws SubSubException { } }  
9.  
10. class CC3 extends CC { void doStuff() throws Exception { } }  
11.  
12. class CC4 extends CC { void doStuff(int x) throws Exception { } }  
13.  
14. class CC5 extends CC { void doStuff()   
      { } }  
  
What is the result? (Choose all that apply.)

*Single Choice - Select one correct answer from the options list.*

Compilation succeeds

Compilation fails due to an error on line 8

Compilation fails due to an error on line 10

Compilation fails due to an error on line 12

Output:

**Q30**

Which of the below statements true?

*Multiple Choice - This may have multiple correct answers. Select required answer(s) from the options list.*

Collections can hold Objects and primitive values.

The synax List<String>list= new ArrayList<>();  
is correct.

next() method returns the next element and does not move the Iterator forward.

hasNext() method determines if more elements exist;

Output:

**Q31**

What will the method length() in the class File return?

*Single Choice - Select one correct answer from the options list.*

The number of characters in the file.

The number of kilobytes in the file.

The number of lines in the file.

None od the above.

Output:

**Q32**

public class Demo {  
// (1) INSERT DECLARATION for accounts variable HERE  
public long getNum(String name)   
{  
Long number = accounts.get(name);  
return number == null ? 0 : number;  
}  
public void setNum(String name, long number)   
{  
accounts.put(name, number);  
}  
}  
  
Select the one correct answer.

*Single Choice - Select one correct answer from the options list.*

private Map<String, long> accounts = new HashMap<String, long>();

private Map accounts = new HashMap();

private Map<String, Long> accounts = new Map<String, Long>();

private Map<String, Long> accounts = new HashMap<String, Long>();

Output:

**Q33**

public class NumberObjectDemo {  
      public static void main(String[] args)   
      {  
        try {  
            NumberObject number1 = new NumberObject();  
        }   
        catch (Throwable t)   
        {  
            t.printStackTrace();  
        }  
        NumberObject number2 = new NumberObject();  
     }  
  }  
  public class NumberObject {  
    static int number = 1 / 0;  
    public NumberObject ()   
   {  
         System.out.println("number undefined");  
    }  
  }

*Single Choice - Select one correct answer from the options list.*

Throws ExceptionInInitializerError and ClassNotFoundException

Gives Output : number undefined

Throws ExceptionInInitializerError and NoClassDefFoundError

Compile Error

Output:

**Q34**

Consider the following code snippet:  
  
Line-1    interface InfA {  
Line-2     protected String getName();  
Line-3    }  
Line-4    public class Test implements InfA{  
Line-5     public String getName()  
                  {  
Line-6        return "test-name";}  
Line-7     public static void main (String[] args)  
                   {  
Line-8      Test t = new Test();  
Line-9      System.out.println(t.getName());  
Line-10     }}  
  
What will happen if above code is executed?

*Single Choice - Select one correct answer from the options list.*

test-name

Compilation fails due to an error on Line 2

Compilation fails due to an error on Line 1

Compilation succeed but Runtime Exception

Output:

**Q35**

class Person{  
 private String name;  
 private int age;  
//Default and parameterized constructor  
//Getter and Setter methods   
}  
  
public class TestCore {  
 public static void main(String[] args)   
{  
 ArrayList<Person> plist = new ArrayList<Person>();  
 plist.add(new Person("Anil",45));  
 plist.add(new Person("Swati",28));  
 plist.add(new Person("Manisha",35));  
 //Line 1  
 }  
}  
  
Refer to the above code of Person and TestCore class, which of the following statement   
will be placed at Line 1 to concatenate all names persons in the list?

*Single Choice - Select one correct answer from the options list.*

System.out.println(plist.stream().min((p,p1)->p.getAge()-p1.getAge()).get().getName());

System.out.println(plist.stream().map(p->p.getAge()).min((a,b)->Integer.compare(a, b)).get());

System.out.println(plist.stream().map(p->p.getAge()).reduce((p,p1)->p+p1).get());

System.out.println(plist.stream().map(p->p.getName()).reduce((p,p1)->p.concat(p1)).get());

Output:

**Q36**

Which of the following functional interface represents an operation upon two operands of the same type  
producing a result of the same type as the operands?

*Single Choice - Select one correct answer from the options list.*

BiFunction

Supplier

BinaryOperator

Consumer

Output:

**Q37**

Which of the following is correct?

*Single Choice - Select one correct answer from the options list.*

ArrayList :-Fast iteration and fast random access  
HashMap :-Fastest updates (key/values); allows one null key, many null  
values.  
Hashtable :-Like a slower HashMap (as with Vector, due to its synchronized  
methods). No null values or null keys allowed  
TreeMap :- A sorted map  
TreeSet :-No duplicates; iterates in sorted order.

Vector :-Fast iteration and fast random access but not synchronized.  
HashMap :-Fastest updates (key/values); allows no null key, many null  
values.  
Hashtable :-Like a slower HashMap, No null values or null keys allowed  
TreeMap:- A sorted map allows duplicates  
TreeSet:- duplicates allowed; iterates in sorted order.

ArrayList :-Fast iteration and fast random access  
HashMap :-Fastest updates (key/values); allows one null key, many null  
values.  
HashSet :-Like a slower HashMap (as with Vector, due to its synchronized  
methods). No null values or null keys allowed  
TreeMap :- A sorted map with duplicate keys and one null.  
TreeSet :-No duplicates; iterates in sorted order.

ArrayList :-Fast iteration and fast random access  
TreeMap :-Fastest updates (key/values); allows one null key, many null  
values.  
HashMap :-Like a slower TreeMap (as with Vector, due to its synchronized  
methods). No null values or null keys allowed  
TreeMap :- unsorted map with ordered selection of elements  
TreeSet No duplicates; iterates in sorted order.

Output:

**Q38**

What will be the result of attempting to compile and run the following code?  
  
public class Demo {  
public static void main(String[] args)   
{  
Set set = new TreeSet<String>();  
set.add("one");  
set.add(2);  
set.add("three");  
System.out.println(set);  
}  
}  
  
Select the one correct answer.

*Single Choice - Select one correct answer from the options list.*

The program does not compile

The program compiles with warning and prints the elements in the set.

The program compiles with warning and throws runtime exception

The program compiles with warning and prints elements of the set.

Output:

**Q39**

List<String> list = new ArrayList<>();  //Line 1  
list.add("Hello");  
list.add("World");  
Consumer<List<String>> consumer = Collections ::reverse; //Line 2  
consumer.accept(list);      
System.out.println(list);  
  
Refer to the above code snippet; what will be the output in console?

*Single Choice - Select one correct answer from the options list.*

Display [World, Hello] in console

Display [Hello, World] in console

Compilation error at Line number 1, as generics not specified

Compilation error at Line number 2, as reverse method returns does not return void

Output:

**Q40**

Given:  
  
2. public class Bunnies {  
3. static int count = 0;  
4. Bunnies()   
    {  
5. while(count < 10) new Bunnies(++count);  
6. }  
7. Bunnies(int x)   
    { super(); }  
8. public static void main(String[] args)   
    {  
9. new Bunnies();  
10. new Bunnies(count);  
11. System.out.println(count++);  
12. }  
13. }  
  
What is the result?

*Single Choice - Select one correct answer from the options list.*

9

10

11

12

compilation error.

Output: