Oracle Certified Professional, Java SE 7 Programmer II Exam

Version 201312

joseluis.iturbide@gmail.com

Question 1

What will be the output when the following program is compiled and run?

```
public class TestClass extends Thread
{
  String name = "";
  public TestClass(String str)
  {
    name = str;
  }
  public void run()
  {
   try
   {
     Thread.sleep( (int) (Math.random()*1000) );
     System.out.println(name);
   }
   catch(Exception e)
   {
   }
  }
  public static void main(String[] str) throws Exception
  {
    Thread t1 = new TestClass("tom");
    Thread t2 = new TestClass("dick");
    t1.start();t2.start();
    t1.join(); t2.join();
    System.out.println("harry");
  }
}
```

Select 1 option(s)

a) It will always print `tom, dick`, and `harry`, in that order.
b) It will always print `harry` in the end.
c) It may print `tom, dick`, and `harry`, in any order.
d) `tom` will always be printed before dick.
e) `tom` will always be printed first.

Question 2

Given the following program, which one of these statements is true?

```
public class TestClass extends Thread
{
   static Object lock1 = new Object();
   static Object lock2 = new Object();
   static volatile int i1, i2, j1, j2, k1, k2;
   public void run()
   {
      while (true)
      {
         workWithLocks();
         workWithoutLocks();
      }
   }
   void workWithLocks()
   {
      synchronized(lock1) { i1++ ;  i2++; }
      synchronized(lock2) { k1++ ; k2++ ; }
      j1++;    j2++;
   }
   void workWithoutLocks()
   {
      if (i1 != i2) System.out.println("i");
      if (j1 != j2) System.out.println("j");
      if (k1 != k2) System.out.println("k");
   }
   public static void main(String args[])
   {
      new TestClass().start();
      new TestClass().start();
   }
}
```

Select 1 option(s)

a) The program will fail to compile.
b) One cannot be certain whether any of the letters i, j and k will be printed during execution.
c) One can be certain that none of the letters i, j and k will ever be printed during execution.
d) One can be certain that the letters i and k will never be printed during execution.
e) One can be certain that the letter k will never be printed during execution.

Question 3

Given :

```
class MyStringComparator implements Comparator
{
    public int compare(Object o1, Object o2)
    {
      int s1 = ((String) o1).length();
      int s2 = ((String) o2).length();
      return s1 - s2;
    }
}
```

and

```
static String[] sa = { "d", "bbb", "aaaa" };
```

Select correct statements.

Select 2 option(s)

a) This is not a valid Comparator implementation.
b) `Arrays.binarySearch(sa, "cc", new MyStringComparator());` will return -2.
c) `Arrays.binarySearch(sa, "c", new MyStringComparator());` will return 0.
d) `Arrays.binarySearch(sa, "c", new MyStringComparator());` will return -1.
e) `Arrays.binarySearch(sa, "c", new MyStringComparator());` will throw an exception.

Question 4

Given:

```
class Game{ }
class Cricket extends Game{ }
class Instrument{ }
class Guitar extends Instrument{ }

interface Player<E>{ void play(E e); }
interface GamePlayer<E extends Game> extends Player<E>{ }
interface MusicPlayer<E extends Instrument> extends Player{ }
```

Identify valid declarations.

Select 1 option(s)

a)
```
class Batsman implements GamePlayer<Cricket>{
    public void play(Game o){ }
}
```
b)
```
class Bowler implements GamePlayer<Guitar>{
    public void play(Guitar o){ }
}
```
c)
```
class Bowler implements Player<Guitar>{
    public void play(Guitar o){ }
}
```
d)
```
class MidiPlayer implements MusicPlayer {
    public void play(Guitar g){ }
}
```
e)
```
class MidiPlayer implements MusicPlayer<Instrument> {
    public void play(Guitar g){ }
}
```

Question 5

Which of the following statements are correct?

Select 1 option(s)

a) Assertions can be enabled or disabled on a class by class basis.
b) Assertions are appropriate to check whether method parameters are valid.
c) Conditional compilation is used to allow an application that uses assertions to run with maximum performance.
d) When an assertion fails, a programmer may either throw an exception or simply return from the method.

Question 6

Which of the given statements will compile and run without exceptions?

Select 5 option(s)

```
a) System.out.printf("\"%-5c\"", 'c');
b) System.out.printf("\"%+0d\""    ,  100);
c) System.out.printf("\"$%(,9.2f\"", -1222.2);
d) System.out.printf("%b", 100);
e) System.out.printf("\"%c\"", new Character('d'));
f) System.out.printf("%s", new Object());
g) System.out.printf("\"%f\"", -100);
```

Question 7

Consider the following code:

```
import java.io.*;
import java.text.*;
import java.util.*;

class ScannerTester
{
    public static void main(String[] args) throws Exception
    {
    FileInputStream fis = new FileInputStream(args[0]);
    Scanner sc = new Scanner(fis);

    //1 insert code here.

    }
}
```

This code is to be used to process a CSV (comma separated values) file that contains decimal numbers and other non-numeric strings separated by comma. What should be inserted at //1 so that it will print out all and only the decimal numbers in the file?

Select 1 option(s)

a)
```
while(sc.hasNext())
   if(sc.hasNextDouble()) System.out.println( sc.nextDouble() );
```

b)
```
sc.useDelimiter(",");
while(sc.hasNext())
  if(sc.hasNextDouble())
    System.out.println( sc.nextDouble() );
```

c)
```
sc.useDelimiter(",");
while(sc.hasNext())
  if(sc.hasNextDouble())
    System.out.println( sc.nextDouble() );
  else sc.next();
```

d)

```
sc.useDelimiter(",");
while(sc.hasNextDouble())
  System.out.println( sc.nextDouble() );
```

e)

```
while(sc.hasNext())
  if(sc.hasNextDouble())
    System.out.println( sc.nextDouble() );
  else sc.next();
```

Question 8

Consider the following program:

```
import java.io.FileReader;
import java.io.FileWriter;

public class ClosingTest {
    public static void main(String[] args) throws Exception {
        try(FileReader fr = new FileReader("c:\\temp\\license.txt");
            FileWriter fw = new FileWriter("c:\\temp\\license2.txt")
)
        {
            int x = -1;
            while( (x = fr.read()) != -1){
                fw.write(x);
            }
        }
    }
}
```

Identify the correct statements.

Select 1 option(s)

a) The FileWriter object will always be closed before the FileReader object.
b) The order of the closure of the FileWriter and FileReader objects is platform dependent and should not be relied upon.
c) The FileWriter object will not be closed if an exception is thrown while closing the FileReader object.
d) This is not a fail safe approach to managing resources because in certain situations one or both of the resources may be left open after the end of the try block.

Question 9

What will the following code fragment print when compiled and run?

```
Statement stmt = null;
Connection c =
DriverManager.getConnection("jdbc:derby://localhost:1527/sample",
"app", "app");
try(stmt = c.createStatement();){
    ResultSet rs = stmt.executeQuery("select * from STUDENT");
    while(rs.next()){
        System.out.println(rs.getString(0));
    }
}catch(SQLException e){
    System.out.println("Exception ");
}
```

(Assume that items not specified such as import statements and try/catch block are all valid.)

Select 1 option(s)

a) It will throw an exception if the first column of the result is not a String.
b) It will throw an exception every time it is run irrespective of what the query returns.
c) It will print the values for the first column of the result and if there is no row in STUDENT table, it will not print anything.
d) It will not compile.

Question 10

Complete the following code by inserting declaration for stateCitiesMap:

```
//Insert line of code here

 List<String> cities = new ArrayList<>();
 cities.add("New York");
 cities.add("Albany");
 stateCitiesMap.put("NY", cities);
```

Select 2 option(s)

a) `Map<String, ArrayList<String>> stateCitiesMap = new HashMap<>();`
b) `Map<String, List<String>> stateCitiesMap =` new HashMap<String, List<>>();
c) `Map<String , ArrayList<String>> stateCitiesMap = new HashMap<<>, List<>>();`

```
d) Map<String, List<String>> stateCitiesMap = new HashMap<String,
   ArrayList<String>>();
e) Map<String, List<String>> stateCitiesMap = new HashMap<>();
f) Map<String, List<String>> stateCitiesMap = new HashMap<String,
   List<String>>();
```

Question 11

Which of the following are characteristics of a class that implements the Singleton pattern?

Select 2 option(s)

a) The class is private.
b) The class has no constructor.
c) The class has one public static method that returns an instance of that class.
d) The class has a private class variable that refers to an instance of the same class.
e) The constructor of the class resolves to the same object each time it is invoked.
f) The class implements Singleton interface.

Question 12

Consider the following class:

```
public class Student {
    private int id;
    private String name;

    public int getId() { return id;  }
    public String getName() { return name;    }
    public void setName(String str) { ... }
    public Student find(int id){ ... }
    public void save(Student s){ ... }
    public void remove(int id){ ... }
    public void update(Student s){ ... }
}
```

What do you need to do to apply the DAO pattern to this class?

Select 2 option(s)

a) Move get/setName and getId methods to another class.
b) Move find, save, remove, and update methods to another class.
c) Create an interface that contains the signatures of find, save, remove, and update methods.
d) Make find, save, remove, and update methods private.
e) Make find, save, remove, and update methods protected.

Question 13

You want to execute a task that returns a result without blocking. Which of the following classes from java.util.concurrent package will be required to achieve this?

Select 4 option(s)

a) Executor
b) ExecutorService
c) Executors
d) Runnable
e) Callable
f) Future

Question 14

Consider the following code.

```
import java.text.*;
import java.util.*;
public class TestClass {
    public static void main(String[] args) throws Exception {
        Date d = new Date();
        DateFormat df = //1 INSERT CODE HERE

        String s = //2 INSERT CODE HERE

        System.out.println(s);
    }
}
```

What should be inserted at //1 and //2 above so that it will print the date in default date format for the UK Locale?

Select 1 option(s)

a)
`df = DateFormat.getDateInstance(DateFormat.DEFAULT, Locale.UK);`
and
`df.formatDate(d);`

b)

```
df = DateFormat.getDateInstance(Locale.UK);
```
and
```
df.format(d);
```

c)
```
df = DateFormat.getDateInstance(DateFormat.DEFAULT, Locale.UK);
```
and
```
df.format(d);
```

d)
```
df = DateFormat.getInstance(DateFormat.DEFAULT, Locale.UK);
```
and
```
df.format(d);
```

e)
```
df = new DateFormat(DateFormat.DEFAULT, Locale.UK);
```
and
```
df.format(d);
```

Question 15

Consider the following code:

```
    public static boolean isValid(Path p){

        return p.startsWith("temp") && p.endsWith("clients.dat");
    }

    public static void writeData() {
        Path p1 = Paths.get("\\temp\\records");
        Path p2 = p1.resolve("clients.dat");
        System.out.println(p2+" "+isValid(p2));
    }
```

What will be printed when the method `writeData()` is executed?

Select 1 option(s)

a) `\temp\records\clients false`
b) `temp\records\clients.dat false`
c) `\temp\records\clients.dat false`
d) `temp\records\clients.dat true`
e) `clients.dat false`
f) `\clients.dat false`

Question 16

Consider the following code:

```
class Bond {
    String ticker; double coupon; java.util.Date maturity;
}

class Portfolio implements Serializable {
    String accountName;
    Bond[] bonds;
}

public class TestClass {
  public static void main(String[] args) throws Exception{
     Portfolio portfolio = // get portfolio somehow.
     // serialize portfolio
  }
}
```

What can be done so that a Portfolio object can be serialized while preserving the state of the Bond objects contained in Portfolio?

Select 2 option(s)

a) It can be serialized as it is without any modification.
b) Just have Bond class implement Serializable.
c) Just make 'bonds' field in Portfolio transient.
d) Change the type of bonds from Bond[] to ArrayList<Bond> bonds;
e) Make bonds array transient in Portfolio and implement readObject and writeObject methods to read and write the state of Bond objects explicitly.

Question 17

Given the following code:

```
enum Title
{
    MR("Mr. "), MRS("Mrs. "), MS("Ms. ");
    private String title;
    private Title(String s){
      title = s;
    }
    public String format(String first, String last){
      return title+" "+first+" "+last;
    }
}

//INSERT CODE HERE
```

Identify valid code snippets ..
(Assume that Title is accessible wherever required.)

Select 4 option(s)

a)

```
void someMethod() {
  System.out.println(Title.format("Rob", "Miller"));
}
```

b)

```
void someMethod() {
  System.out.println(Title.MR.format("Rob", "Miller"));
}
```

c)

```
class TestClass{
  void someMethod() {
    System.out.println(MR.format("Rob", "Miller"));
  }
}
```

d)

```
enum Title2 extends Title {
  DR("Dr. ");
}
```

e)

```
class TestClass{
  void someMethod()  {
     Title.DR dr = new Title.DR("Dr. ");
  }
}
```

f)

```
enum Title2 {
   DR;
   private Title t;
}
```

g)

```
enum Title2 {
   DR;
   private Title t = Title.MR;
}
```

h)

```
enum Title2 {
   DR;
   private Title t = Title.MR;
   public String format(String s){ return t.format(s, s); };
}
```

Question 18

Given:

Class A has a reference to an object of class X
Class B is-a A
A calls public instance methods of class Y
B accesses public instance fields of class Z

Which of the following statements are correct?

Select 3 option(s)

a) Class B has high cohesion.
b) Level of cohesion of A and B cannot be determined.
c) Z violates encapsulation.
d) B violates encapsulation.
e) A is loosely coupled to Y
f) B has-a Y

Question 19

Given the following code, which of the constructors given in the options can be added to class B without causing a compile time error?

```
class A
{
   int i;
   public A(int x) { this.i = x; }
}
class B extends A
{
   int j;
   public B(int x, int y) { super(x);  this.j = y; }
}
```

Select 2 option(s)

a) `B( ) { }`
b) `B(int y ) { j = y; }`
c) `B(int y ) { super(y*2 ); j = y; }`
d) `B(int y ) { i = y; j = y*2; }`
e) `B(int z ) { this(z, z); }`

Question 20

Consider the following class:

```
public class GoodOne {
   int theval;
   public int hashCode() {
      return theval%3;
   }
   public boolean equals(Object obj) {
      try{
      // 1 insert code here.
      }catch(Exception e) {
        return false;
      }
   }
}
```

Which of the following options may be inserted at //1?

Select 1 option(s)

a) `return true;`
b) `return this == obj? true : (theval%3 == 0 && ((GoodOne)obj).theval%3==0) ? true :false;`
c) `return theval%2 == 0? true :false;`
d) `return ( (int)Math.random())*10%3 == 0? true :false;`
   Assume that `Math.random()` returns a double between 0.0 and 1.0 (not including 1.0).
e) `return false;`

Question 21

What will the following code print when compiled and run?

```
public class TestSIS {

    public static void main(String[] args) {
        String str = null;
        switch(str){
            case "null" : System.out.println("1");
                break;
            case "" : System.out.println("2");
                break;
            default : System.out.println("3");
        }
    }
}
```

Select 1 option(s)

a) 1
b) 2
c) 3
d) a NullPointerException stack trace
e) The code will not compile.
f) It will not print anything.

Question 22

Which of the following are valid enum values defined in `java.nio.file.FileVisitResults`?

Select 2 option(s)

a) SKIP_SIBLINGS
b) CONTINUE_SIBLINGS
c) SKIP_SUBTREE
d) SKIP_TREE
e) SKIP

Question 23

Consider the following line of code appearing an application:

```
System.out.println("Student is "+student);
```

where `student` refers to an object of class `somepackage.Student`.
The intention is to print the `id` and `name` attributes of the `Student` object but it actually prints some thing like `somepackage.Student@97a560`.
What is wrong?

Select 1 option(s)

a) The object referred to by `student` is uninitialized.
b) Student class does not have `id` and `name` attributes.
c) `Student` class is not accessible where this line of appears.
d) `Student` class does not properly override `toString()` method.

Question 24

Assume that the following directory exists:

```
c:\a\b\c
```

A File object is created as follows:

```
File f = new File("c:\\a\\b\\c\\d\\e");
```

Given that directories `d` and `e` do not exist under `c`, which of the following statements are correct?

Select 2 option(s)

a) The given line of code will throw an exception at run time.
b) `f.mkdir();` will create directory `d` under `c` and directory `e` under `d`.
c) `f.mkdirs();` will create directory `d` under `c` and directory `e` under `d`.
d) `f.getParentFile()` will return a File Object representing `c:\a\b\c\d`
e) None of these.

Question 25

Which of the following RowSets always maintains a connection to the database?

Select 1 option(s)

a) FilteredRowSet
b) ConnectedRowSet
c) WebRowSet
d) JoinRowSet
e) JdbcRowSet