➢ Vendor: Oracle

➢ Exam Code: 1Z0-808

➢ Exam Name: Java SE 8 Programmer I

➢ Question 61 -- Question 80

QUESTION 61
Which statement best describes encapsulation?

A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.
B. Encapsulation ensures that classes can be designed so that their methods are inheritable.
C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.
D. Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

Answer: A
Explanation:
http://www.tutorialspoint.com/java/java_encapsulation.htm

QUESTION 62
Given the code fragment from three files:

SalesMan.java:

```
package sales;
public class SalesMan { }
```

Product.java:

```
package sales.products;
public class Product { }
```

Market.java:

```
1. package market;
2. // insert code here
3. public class USMarket {
4.     SalesMan sm;
5.     Product p;
6. }
```

Which code fragment, when inserted at line 2, enables the code to compile?

A)
```
import sales.*;
```

B)
```
import java.sales.products.*;
```

C)
```
import sales;
import sales.products;
```

D)
```
import sales.*;
import products.*;
```

E)
```
import sales.*;
import sales.products.*;
```

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: E
Explanation:
https://docs.oracle.com/javase/tutorial/java/package/usepkgs.html

QUESTION 63
Given the following class:

```
public class CheckingAccount {
    public int amount;
    public CheckingAccount(int amount){
        this.amount = amount;
    }
    public int getAmount(){
        return amount;
    }
    public void changeAmount(int x){
        amount += x;
    }
}
```

And given the following main method, located in another class:

```
public static void main(String[] args) {
    CheckingAccount acct = new CheckingAccount((int)(Math.random()*1000));
    //line n1
    System.out.println(acct.getAmount());
}
```

Which three lines, when inserted independently at line n1, cause the program to print a o balance?

A. this.amount = 0;
B. amount = 0;
C. acct (0) ;
D. acct.amount = 0;
E. acct. getAmount () = 0;
F. acct.changeAmount(0);
G. acct.changeAmount(-acct.amount);
H. acct.changeAmount(-acct.getAmount());

Answer: DGH
Explanation:
A and B don't compile because there isn't a variable amount in method main.
C is wrong because we can't call the constructor acct directly.
E is wrong because we can't make a method on acct equal to 0.
F is wrong because does not change variable amount of class CheckingAccount.

QUESTION 64
Given the code fragment:

```
String shirts[][] = new String[2][2];
shirts[0][0] = "red";
shirts[0][1] = "blue";
shirts[1][0] = "small";
shirts[1][1] = "medium";
```

Which code fragment prints red: blue: small: medium?

A)
```
for (int index = 1; index < 2; index++) {
    for (int idx = 1; idx < 2; idx++) {
        System.out.print(shirts[index][idx] + ":");
    }
}
```

B)
```
for (int index = 0; index < 2; ++index) {
    for (int idx = 0; idx < index; ++idx) {
        System.out.print(shirts[index][idx] + ":");
    }
}
```

C)
```
for (String c : colors) {
    for (String s : sizes) {
        System.out.println(s + ":");
    }
}
```

D)
```
for (int index = 0; index < 2;) {
    for (int idx = 0; idx < 2;) {
        System.out.print(shirts[index][idx] + ":");
        idx++;
    }
    index++;
}
```

A. Option A
B. Option B
C. Option C
D. Option D

Answer: B

QUESTION 65
Given the code fragment:

```
public class Test{

    void readCard(int cardNo) throws Exception {
        System.out.println("Reading Card");
    }

    void checkCard(int cardNo) throws RuntimeException { // line n1
        System.out.println("Checking Card");
    }

    public static void main(String[] args) {
        Test ex = new Test();
        int cardNo = 12344;
        ex.checkCard(cardNo);                    //line n2
        ex.readCard(cardNo);                     //line n3
    }
}
```

What is the result?

A. Reading Card
   Checking Card
B. Compilation fails only at line n1.
C. Compilation fails only at line n2.
D. Compilation fails only at line n3.
E. Compilation fails at both line n2 and line n3.

Answer: D
Explanation:
Exception is a checked exception so we are required to check it with try/catch or be declared in method main.

QUESTION 66
Given the code fragment:

```
public static void main(String[] args) {
    StringBuilder sb = new StringBuilder(5);
    String s = "";

    if (sb.equals(s)) {
        System.out.println("Match 1");
    } else if (sb.toString().equals(s.toString())) {
        System.out.println("Match 2");
    } else {
        System.out.println("No Match");
    }
}
```

What is the result?

A. Match 1
B. Match 2
C. No Match

D. A NullPointerException is thrown at runtime.

Answer: B

QUESTION 67
Given:

```
package p1;
public class Acc {
    int p;
    private int q;
    protected int r;
    public int s;
}
```

Test.java:

```
package p2;
import p1.Acc;
public class Test extends Acc {
    public static void main(String[] args) {
        Acc obj = new Test();
    }
}
```

Which statement is true?

A. Both p and s are accessible by obj.
B. Only s is accessible by obj.
C. Both r and s are accessible by obj.
D. p, r, and s are accessible by obj.

Answer: B
Explanation:
Only s is accessible because it is the only public member of class Acc.

QUESTION 68
Given:

Base.java:

```
class Base {
    public void test(){
        System.out.println("Base ");
    }
}
```

DerivedA.java:

```
class DerivedA extends Base {
    public void test(){
        System.out.println("DerivedA ");
    }
}
```

DerivedB.java:

```
class DerivedB  extends DerivedA {
    public void test(){
        System.out.println("DerivedB ");
    }
    public static void main(String[] args) {
        Base b1 = new DerivedB();
        Base b2 = new DerivedA();
        Base b3 = new DerivedB();
        b1 = (Base) b3;
        Base b4 = (DerivedA) b3;
        b1.test();
        b4.test();
    }
}
```

What is the result?

A. Base
DerivedA
B. Base
DerivedB
C. DerivedB
DerivedB
D. DerivedB
DerivedA
E. A classcast Except ion is thrown at runtime.

Answer: C

QUESTION 69
Given the code fragment:

```
public static void main(String[] args) {
    ArrayList myList = new ArrayList();
    String[] myArray;
    try {
       while (true) {
           myList.add("My String");
       }
    }
    catch (RuntimeException re) {
       System.out.println("Caught a RuntimeException");
    }
    catch (Exception e) {
       System.out.println("Caught an Exception");
    }
    System.out.println("Ready to use");
}
```

What is the result?

A. Execution terminates in the first catch statement, and caught a RuntimeException is printed to the console.
B. Execution terminates In the second catch statement, and caught an Exception is printed to the console.
C. A runtime error is thrown in the thread "main".
D. Execution completes normally, and Ready to use is printed to the console.
E. The code fails to compile because a throws keyword is required.

Answer: C
Explanation:
while loop is an infinite loop so the program ends with an OutOfMemoryError.
This error can't be caught with Exception nor RuntimeException.
http://stackoverflow.com/questions/1692230/is-it-possible-to-catch-out-of-memory-exception-in-java

QUESTION 70
Given:

```
System.out.println("5 + 2 = " + 3 + 4);
System.out.println("5 + 2 = " + (3 + 4));
```

What is the result?

A)
```
5 + 2 = 34
5 + 2 = 34
```

B)
```
5 + 2 + 3 + 4
5 + 2 = 7
```

C)
```
7 = 7
7 + 7
```

D)
```
5 + 2 = 34
5 + 2 = 7
```

A. Option A
B. Option B
C. Option C
D. Option D

Answer: D
Explanation:
If neither operand of + is a reference to a String object, the operator is the arithmetic addition operator, not the string concatenation operator. Note that Java does not allow a program to define overloaded operators. However, the language defines the + operator to have a meaning that is fundamentally different from arithmetic addition if at least one of its operands is a String object.
The way in which Java decides if + means arithmetic addition or string concatenation means that the use of parentheses can alter the meaning of the + operator.
See "String Concatenation Operator +" at
http://oponet.stsci.edu/web/documentation/Java%20Reference%20Library%201.02/langref/ch04_06.htm

QUESTION 71
Given:

```
public static void main(String[] args) {
    String ta = "A ";
    ta = ta.concat("B ");
    String tb = "C ";
    ta = ta.concat(tb);
    ta.replace('C', 'D');
    ta = ta.concat(tb);
    System.out.println(ta);
}
```

What is the result?

A. A B C D
B. A C D
C. A B C C
D. A B D
E. A B D C

Answer: C
Explanation:
The line "ta.replace('C', 'D');" returns a string that is never assigned to ta.

QUESTION 72
Given the code fragment:

```
3. public static void main(String[] args) {
4.       int x = 5;
5.       while (isAvailable(x)) {
6.           System.out.print(x);
7.
8.       }
9. }
10.
11. public static boolean isAvailable(int x) {
12.     return x-- > 0 ? true : false;
13. }
```

Which modification enables the code to print 54321?

A. Replace line 6 with System, out. print (--x) ;
B. At line 1, insert x --;
C. Replace line 6 with --x; and, at line 7, insert system, out. print (x);
D. Replace line 12 With return (x > 0) ? false: true;

Answer: A

QUESTION 73
Given the code fragment:

```
4. public static void main(String[] args) {
5.     boolean opt = true;
6.     switch (opt) {
7.         case true:
8.             System.out.print("True");
9.             break;
10.        default:
11.            System.out.print("***");
12.    }
13.    System.out.println("Done");
14. }
```

Which modification enables the code fragment to print TrueDone?

A. Replace line 5 With String result = "true";
   Replace line 7 with case "true":
B. Replace line 5 with boolean opt = l;
   Replace line 7 with case 1=
C. At line 9, remove the break statement.
D. Remove the default section.

Answer: A
Explanation:
Switch statements with String cases were implemented in Java SE 7.

QUESTION 74
Given the following main method:

```
public static void main(String[] args) {
    int num = 5;
    do {
        System.out.print(num-- +" ");
    } while(num == 0);
}
```

What is the result?

A. 5 4 3 2 1 0
B. 5 4 3 2 1
C. 4 2 1
D. 5
E. Nothing is printed

Answer: D
Explanation:
The loop body executes only once because on the while condition num = 4.
When the execution reaches System.out.print, num = 5.

QUESTION 75
Given the code fragment:

```
int x = 100;
int a = x++;
int b = ++x;
int c = x++;
int d = (a < b) ? (a < c) ? a: (b <c )? b: c;
System.out.println(d);
```

What is the result?

A. 100
B. 101
C. 102
D. 103
E. Compilation fails

Answer: E
Explanation:
Compilation fails with error ": expected" because we have three ternary operators but only two colons.

QUESTION 76
Given:

```
public class Test {

    public static void main(String[] args) {

        String[][] chs = new String[2][];
        chs[0] = new String[2];
        chs[1] = new String[5];
        int i = 97;

        for (int a = 0; a < chs.length; a++) {
            for (int b = 0; b < chs.length; b++) {
                chs[a][b] = "" + i;
                i++;
            }
        }

        for (String[] ca : chs) {
            for (String c : ca) {
                System.out.print(c + " ");
            }
            System.out.println();
        }
    }
}
```

What is the result?

A. 97 98
99 100 null null null
B. 91 98
99 100 101 102 103
C. Compilation rails.
D. A NullPointerException is thrown at runtime.
E. An ArrayIndexOutOfBoundsException is thrown at runtime.

Answer: A
Explanation:
When we exit first loop we have
chs[0][0] = 97
chs[0][1] = 98
chs[1][0] = 99
chs[1][1] = 100
chs[1][2] = null;
chs[1][3] = null;
chs[1][4] = null;
The second loop prints these values.

QUESTION 77
Given the code fragment:

```
public class Employee {
    String name;
    boolean contract;
    double salary;
    Employee() {
        // line n1
    }
    public String toString(){
        return name + ":" + contract + ":" + salary;
    }
    public static void main(String[] args) {
        Employee e = new Employee();
        // line n2
        System.out.print(e);
    }
}
```

Which two modifications, when made independently, enable the code to print joe:true: 100.0?

☐ A) Replace `line n2` with:

```
e.name = "Joe";
e.contract = true;
e.salary = 100;
```

☐ B) Replace `line n2` with:

```
this.name = "Joe";
this.contract = true;
this.salary = 100;
```

☐ C) Replace `line n1` with:

```
this.name = new String("Joe");
this.contract = new Boolean(true);
this.salary = new Double(100);
```

☐ D) Replace `line n1` with:

```
name = "Joe";
contract = TRUE;
salary = 100.0f;
```

☐ E) Replace `line n1` with:

```
this("Joe", true, 100);
```

A. Option A
B. Option B
C. Option C
D. Option D
E. Option E

Answer: AC

QUESTION 78
View the exhibit:

```
public class Student {
public String name = "";
public int age = 0;
public String major = "Undeclared";
public boolean fulltime = true;
public void display() {
System.out.println("Name: " + name + " Major: " + major); }
public boolean isFullTime() {
return fulltime;
}
}
```

Which line of code initializes a student instance?

A. Student student1;
B. Student student1 = Student.new();
C. Student student1 = new Student();
D. Student student1 = Student();

Answer: C

QUESTION 79
What should keyword1 and keyword2 be respectively, in oreder to produce output 2345?

```
int [] array = {1,2,3,4,5};
for (int i: array) {
if ( i < 2) {
keyword1 ;
}
System.out.println(i);
if ( i == 3) {
keyword2 ;
}}
```

A. continue, break
B. break, break
C. break, continue
D. continue, continue

Answer: D

QUESTION 80
What is the result?

```
int i, j=0;
i = (3* 2 +4 +5 ) ;
j = (3 * ((2+4) + 5));
System.out.println("i:"+ i + "\nj":+j);
```

A. i: 16
j: 33
B. i: 15
j: 33
C. i: 33
j: 23
D. i: 15
j: 23

A. Option A
B. Option B
C. Option A
D. Option D

Answer: B