Java Software Development Quiz 7

1. Which of these array declaration statements are not legal?

Select the two correct answers.

(a) int[] i[] = { { 1, 2 }, { 1 }, {}, { 1, 2, 3 } };

(b) int i[] = new int[2] {1, 2};

(c) int i[][] = new int[][] { {1, 2, 3}, {4, 5, 6} };

(d) int i[][] = { { 1, 2 }, new int[2] };

(e) int i[4] = { 1, 2, 3, 4 };

2. What will be the result of compiling and running the following program?

```
public class Passing {
   public static void main(String[] args) {
      int a = 0; int b = 0;
      int[] bArr = new int[1]; bArr[0] = b;
      inc1(a); inc2(bArr);
      System.out.println("a=" + a + " b=" + b + " bArr[0]=" + bArr[0]);
   }
   public static void inc1(int x) { x++; }
   public static void inc2(int[] x) { x[0]++; }
}
```

Select the one correct answer.

(a) The code will fail to compile, since x[0]++; is not a legal statement.

(b) The code will compile and will print "a=1 b=1 bArr[0]=1", when run.

(c) The code will compile and will print "a=0 b=1 bArr[0]=1", when run.

(d) The code will compile and will print "a=0 b=0 bArr[0]=1", when run.

(e) The code will compile and will print "a=0 b=0 bArr[0]=0", when run.

3. Which method declarations are valid declarations? (課外補充，考試不會考)

Select the three correct answers.

(a) void compute(int... is) { }

(b) void compute(int is...) { }

(c) void compute(int... is, int i, String... ss) { }

(d) void compute(String... ds) { }

(e) void compute(String... ss, int len) { }

(f) void compute(char[] ca, int... is) { }

4. Given the class

```
// File name: Args.java
public class Args {
  public static void main(String[] args) {
    System.out.println(args[0] + " " + args[args.length-1]);
  }
}
```

what would be the result of executing the following command line?

```
   >java Args In politics stupidity is not a handicap
```

Select the one correct answer.

(a) The program will throw an ArrayIndexOutOfBoundsException.

(b) The program will print "java handicap".

(c) The program will print "Args handicap".

(d) The program will print "In handicap".

(e) The program will print "Args a".

(f) The program will print "In a".

5. Given the following class, which of these alternatives are valid ways of referring to the class from outside of the package net.basemaster?

```
package net.basemaster;
public class Base {
   // ...
}
```

(a) By simply referring to the class as Base.

(b) By simply referring to the class as basemaster.Base.

(c) By simply referring to the class as net.basemaster.Base.

(d) By importing with net.basemaster.*, and referring to the class as Base.

(e) By importing with net.*, and referring to the class as basemaster.Base.

Answer

1. (b) and (e)

The size of the array cannot be specified, as in (b) and (e). The size of the array is given implicitly by the initialization code. The size of the array is never specified in the declaration of an array reference. The size of an array is always associated with the array instance (on the right-hand side), not the array reference (on the left-hand side).

2. (d)

The variables a and b are local variables that contain primitive values. When these variables are passed as arguments to another method, the method receives copies of the primitive values in the variables. The actual variables are unaffected by operations performed on the copies of the primitive values within the called method. The variable bArr contains a reference value that denotes an array object containing primitive values. When the variable is passed as a parameter to another method, the method receives a copy of the reference value. Using this reference value, the method can manipulate the object that the reference value denotes. This allows the elements in the array object referenced by bArr to be accessed and modified in the method inc2().

3. (a), (d), and (f)

The ellipses (...) must be specified before the parameter name. Only one varargs parameter is permitted, and it must be the last parameter in the formal parameter list.

4. (d)

The length of the array passed to the main() method is equal to the number of program arguments specified in the command line. Unlike some other programming languages, the element at index 0 does not contain the name of the program. The first argument given is retrieved using args[0], and the last argument given is retrieved using args[args.length-1].

5. (c) and (d)

A class or interface name can be referred to by using either its fully qualified name or its simple name. Using the fully qualified name will always work, but in order to use the simple name it has to be imported. By importing net.basemaster.* all the type names from the package net.basemaster will be imported and can now be referred to using simple names. Importing net.* will not import the subpackage basemaster.