Java Programmer Certification Mock Exam No 3

Last Updated 25 March 2000

56 Questions

This is my mock Exam No 3 based on the Objectives for the Sun Java Programmers Exam. It was created by Marcus Green (mail@marcusgreen.co.uk) and may be freely distributed so long as it is unmodified. Please email me if you have any corrections or comments.

How does this compare with the real thing?

You should assume that the real thing will be harder. I have had reports that people taking the real thing have got a mark within 10% of the mark they get on my earlier.

Does it come with answers?

Yes, this is a long document (around 35 pages), if you cannot see the answers then you have not got the entire document, try reloading it until you can see the answer to the final question and the words "End of document". The answers contain references to approximately which objective the question relates to.

How does this compare with the real thing?

The questions are of a similar format to the real questions, though you get a few "fill in the text box" questions on the real thing. You should assume that the real thing will be harder though many people have told me that they get similar marks in the real thing to my exams.. This one is probably easier than my Exam No 1

Where can you find other Mock Exams?

Check out my FAQ at http://www.jchq.net/faq/jcertfaq.htm for links to other mock exams You can check out my first mock exam that contains 60 question at http://www.jchq.net/mockexams/exam1.htm

Questions

Question 1)

Which of the following are legal statements?

1) float f=1/3;

2) int i=1/3;
3) float f=1.01;
4) double d=999d;

Answer to Question 1)

Question 2)

Which of the following are Java keywords?

1) NULL
2) new
3) instanceOf
4) wend

Answer to Question 2)

Question 3)

Which of the following are valid statements?

1) System.out.println(1+1);
2) int i=2+'2';
3) String s="on"+'one';
4) byte b=255;

Answer to Question 3)

Question 4)

Which of the following statements are true?

1) The garbage collection algorithm in Java is vendor implemented
2) The size of primitives is platform dependent
3) The default type for a numerical literal with decimal component is a float.
4) You can modify the value in an Instance of the Integer class with the setValue method

Answer to Question 4)

Question 5)

Which of the following are true statements?

1) I/O in Java can only be performed using the Listener classes

2) The RandomAccessFile class allows you to move directly to any point a file.
3) The creation of a named instance of the File class creates a matching file in the underlying operating system only when the close method is called.
4) The characteristics of an instance of the File class such as the directory separator, depend on the current underlying operating system

Answer to Question 5)

Question 6).

Which of the following statements are true?

1) The instanceof operator can be used to determine if a reference is an instance of a class, but not an interface.
2) The instanceof operator can be used to determine if a reference is an instance of a particular primitive wrapper class
3) The instanceof operator will only determine if a reference is an instance of a class immediately above in the hierarchy but no further up the inheritance chain
4) The instanceof operator can be used to determine if one reference is of the same class as another reference thus

Answer to Question 6)

Question 7)

Which of the following statements are true?

1) An interface can only contain method and not variables
2) Java does not allow the creation of a reference to an interface with the new keyword.
3) A class may extend only one other class and implement only one interface
4) Interfaces are the Java approach to addressing its single inheritance model, but require implementing classes to create the functionality of the Interfaces.

Answer to Question 7)

Question 8)

Which of the following are valid statements

1) public class MyCalc extends Math
2) Math.max(s);
3) Math.round(9.99,1);
4)Math.mod(4,10);

Answer to Question 8)

Question 9)

Which of the following are methods of the Runnable interface

1) run
2) start
3) yield
4) stop

Answer to Question 9)

Question 10)

Which of the following statements are true?

1) A byte can represent between -128 to 127
2) A byte can represent between -127 to 128
3) A byte can represent between -256 to 256
4) A char can represent between -2x2 pow 16 2 x2 pow 16 - 1

Answer to Question 10)

Question 11)

What will happen when you attempt to compile and run the following code

```
class Base{
public void Base(){
 System.out.println("Base");
 }
}
public class In extends Base{
public static void main(String argv[]){
 In i=new In();
 }
}
```

1) Compile time error Base is a keyword
2) Compilation and no output at runtime
3) Output of Base
4) Runtime error Base has no valid constructor

Answer to Question 11)

Question 12)

You have a public class called myclass with the main method defined as follows

```
public static void main(String parm[]){
    System.out.println(parm[0]);
 }
```

If you attempt to compile the class and run the program as follows

```
java myclass hello
```

What will happen?

1) Compile time error, main is not correctly defined
2) Run time error, main is not correctly defined
3) Compilation and output of java
4) Compilation and output of hello

Answer to Question 12)

Question 13)

Which of the following statements are true?

1) If a class has any abstract methods it must be declared abstract itself.
2) All methods in an abstract class must be declared as abstract
3) When applied to a class, the *final* modifier means it cannot be sub-classed
4) transient and volatile are Java modifiers

Answer to Question 13)

Question 14)

Objective 1.2)

Which of the following are valid methods?

1) public static native void amethod(){}
2) public static void amethod(){}
3) private protected void amethod(){}
4) static native void amethod();

Answer to Question 14)

Question 15)

Objective 6.2)

Which of the following statements are true?

1) Constructors cannot have a visibility modifier
2) Constructors can be marked public and protected, but not private
3) Constructors can only have a primitive return type
4) Constructors are not inherited

Answer to Question 15)

Question 16)

What will happen when you attempt to compile and run the following class?

```
class Base{
        Base(int i){
        System.out.println("Base");
        }
}
class Severn extends Base{
public static void main(String argv[]){
        Severn s = new Severn();
        }
        void Severn(){
        System.out.println("Severn");
        }
}
```

1) Compilation and output of the string "Severn" at runtime
2) Compile time error
3) Compilation and no output at runtime
4) Compilation and output of the string "Base"

Answer to Question 16)

Question 17)

Which of the following statements are true?

1) static methods do not have access to the implicit variable called *this*

2) a static method may not be overriden
3) a static method may not be overriden to be non-static
4) a static method may not be overloaded

Answer to question 17)

Question 18)

Which of the following will compile without error?

1)

```
char c='  1'  ;
System.out.println(c>>1);
```

2)

```
Integer i=Integer("1");
System.out.println(i>>1);
```

3)

```
int i=1;
System.out.println(i<<<1);
```

4)

```
int i=1;
System.out.println(i<<1);
```

Answer to Question 18)

Question 19)

Which of the following are true?

1) A component may have only one event listener attached at a time
2) An event listener may be removed from a component
3) The ActionListener interface has no corresponding Adapter class
4) The processing of an event listener requires a try/catch block

Answer to Question 19)

Question 20)

Which of the following are Java keywords?

1) sizeof
2) main
3) transient
4) volatile

Answer to Question 20)

Question 21)

Which of the following statements are true?

1) The default constructor has a return type of void
2) The default constructor takes a parameter of void
3) The default constructor takes no parameters
4) The default constructor is not created if the class has any constructors of its own.

Answer to Question 21)

Question 22)

Which of the following statements are true?

1) All of the variables in an interface are implicitly static
2) All of the variables in an interface are implicitly final
3) All of the methods in an interface are implicitly abstract
4) A method in an interface can access class level variables

Answer to Question 22)

Question 23)

Which of the following statements are true?

1) The String class is implemented as a char array, elements are addressed using the stringname[] convention
2) The + operator is overloaded for concatenation for the String class

3) Strings are a primitive type in Java and the StringBuffer is used as the matching wrapper type
4) The size of a string can be retrieved using the length property

Answer to Question 23)

Question 24)

Which of the following statements are true?

1) A method in an interface must not have a body
2) A class may extend one other class plus at most one interface
3) A class may extends at most one other class plus implement many interfaces
4) An class accesses an interface via the keyword *uses*

Answer to Question 24)

Question 25)

Which of the following statements are true?

1) The following statement will produce a result of 1. System.out.println(-1 >>>2);
2) Performing an unsigned left shift (<<<) on a negative number will always produce a negative number result
3) The following statement will produce a result of zero, System.out.println(1 >>1);
4) All the integer primitives in java are signed numbers

Answer to Question 25)

Question 26)

Which of the following statements are true?

1) The elements in a Java array can only be of primitive types, not objects
2) Arrays are initialized to default values wherever they are created
3) An array may be dynamically resized using the setSize method
4) You can find out the size of an array using the size method

Answer to Question 26)

Question 27)

Given the following class

```
public class Ombersley{
public static void main(String argv[]){
        boolean b1 = true;
        if((b1 ==true) || place(true)){
                System.out.println("Hello Crowle");
                }
        }
        public static boolean place(boolean location){
        if(location==true){
                System.out.println("Borcetshire");
                }
        System.out.println("Powick");
        return true;
        }
}
```

What will happen when you attempt to compile and run it?

1) Compile time error
2) Output of "Hello Crowle"
3) Output of Hello Crowle followed by Borcetshire and Powick
4) No output

Answer to Question 27)

Question 28)

You are given a class hierarchy with an instance of the class Dog. The class Dog is a child of mammal and the class Mammal is a child of the class Vertibrate. The class Vertibrate has a method called *move* which prints out the string "move". The class mammal overrides this method and prints out the string "walks". The class Dog overrides this method and prints out the string "walks on paws". Given an instance of the class Dog,. how can you access the ancestor method *move* in Vertibrate so it prints out the string "move";

1) d.super().super().move();
2) d.parent().parent().move();
3) d.move();
4) none of the above;

Answer to Question 28)

Question 29)

Which of the following most closely describes the process of overriding?

1) A class with the same name replaces the functionality of a class defined earlier in the hierarchy
2) A method with the same name completely replaces the functionality of a method earlier in the hierarchy

3) A method with the same name but different parameters gives multiple uses for the same method name
4) A class is prevented from accessing methods in its immediate ancestor

Answer to Question 29)

Question 30)

Which of the following statements are true?

1) The % is used to calculate a percentage thus: 10 % 20=50
2) The / operator is used to divide one value by another
3) The # symbol may not be used as the first character of a variable
4) The $ symbol may not be used as the first character of a variable

Answer to Question 30)

Question 31)

Which of the following statements are true?

1) The default layout manager for an Applet is FlowLayout
2) The default layout manager for a Frame is FlowLayout
3) A layout manager must be assigned to an Applet before the setSize method is called
4) The FlowLayout manager attempts to honor the preferred size of any components

Answer to Question 31)

Question 32)

Which of the following statements are true about a variable created with the static modifier?

1) Once assigned the value of a static variable may not be altered
2) A static variable created in a method will keep the same value between calls
3) Only one instance of a static variable will exist for any amount of class instances
4) The static modifier can only be applied to a primitive value

Answer to Question 32)

Question 33)

Which of the following statements are true?

1) Java uses a system called UTF for I/O to support international character sets

2) The RandomAccessFile is the most suitable class for supporting international character sets
3) An instance of FileInputStream may not be chained to an instance of FileOutputStream
4) File I/O activities requires use of Exception handling

Answer to Question 33)

Question 34)

What will happen when you attempt to compile and run the following code?

```
import java.io.*;
class ExBase{
abstract public void martley(){

        }
}
public class MyEx extends ExBase{
        public static void main(String argv[]){
        DataInputStream fi = new DataInputStream(System.in);
        try{
        fi.readChar();
        }catch(IOException e){
        System.exit(0);
        }
        finally {System.out.println("Doing finally");}
        }
}
```

1) Compile time error
2) It will run, wait for a key press and then exit
3) It will run, wait for a keypress, print "Doing finally" then exit
4) At run and immediately exit

Answer to Question 34)

Question 35)

What will happen when you attempt to compile and run the following code

```
public class Borley extends Thread{
public static void main(String argv[]){
        Borley b = new Borley();
        b.start();
        }
public void run(){
        System.out.println("Running");
        }
}
```

1) Compilation and run but no output
2) Compilation and run with the output "Running"
3) Compile time error with complaint of no Thread target

4) Compile time error with complaint of no access to Thread package

Answer to Question 35)

Question 36)

Assuming any exception handling has been set up, which of the following will create an instance of the RandomAccessFile class

1) RandomAccessFile raf=new RandomAccessFile("myfile.txt","rw");
2) RandomAccessFile raf=new RandomAccessFile(new DataInputStream());
3) RandomAccessFile raf=new RandomAccessFile("myfile.txt");
4) RandomAccessFile raf=new RandomAccessFile(new File("myfile.txt"));

Answer to Question 36)

Question 37)

Given the following class definition

```
public class Upton{
public static void main(String argv[]){

        }
        public void amethod(int i){}
        //Here
}
```

Which of the following would be legal to place after the comment //Here ?
1) public int amethod(int z){}
2) public int amethod(int i,int j){return 99;}
3) protected void amethod(long l){ }
4) private void anothermethod(){}

Answer to Question 37)

Question 38)

Is the following statement

1) true or

2) false?

"a nested top-level class cannot declare any *static* variables, *static* methods, or *static* initializers."

Answer to Question 38)

Question 39)

Given the following class definition

```
public class Droitwich{
        class one{
                private class two{
                        public void main(){
                        System.out.println("two");
                        }
                }
        }
}
```

Which of the following statements are true

1) The code will not compile because the classes are nested to more than one level
2) The code will not compile because class two is marked as private
3) The code will compile and output the string two at runtime
4) The code will compile without error

Answer to Question 39)

Question 40)

Given the following code

```
class Base{
static int oak=99;
}
public class Doverdale extends Base{
public static void main(String argv[]){
        Doverdale d = new Doverdale();
        d.amethod();
        }
        public void amethod(){
        //Here
        }
}
```

Which of the following if placed after the comment //Here, will compile and modify the value of the variable oak?

1) super.oak=1;
2) oak=33;
3) Base.oak=22;
4) oak=50.1;

Answer to Question 40)

Question 41)

You are creating an application that has a form with a text entry field used to enter a persons age. Which of the following is appropriate for capturing this information.

1) Use the getText field of a TextField and parse the result using Integer
2) Use the getInteger field of the TextField
3) Use the getText field of a TextBox and parse the result using the getInt method of Integer class
4) Use the getText field of a Textfield and use the parseInt method of the Integer class

Answer to Question 41)

Question 42)

Given the following declaration

Integer i=new Integer(99);

How can you now set the value of i to 10?

1) i=10;
2) i.setValue(10);
3) i.parseInt(10);
4) none of the above

Answer to Question 42)

Question 43)

Which of the following statements are true

1) constructors cannot be overloaded
2) constructors cannot be overridden
3) a constructor can return a primitive or an object reference
4) Constructor invocation occurs from the current class up the hierarchy to the ancestor class

---

Question 44)

Given a reference called

t

to to a class which extends Thread, which of the following will cause it to give up cycles to allow another thread to execute.

1) t.yield();
2) yield()
3) yield(100) //Or some other suitable amount in milliseconds
4) yield(t);

Answer to Question 44)

Question 45)

What will happen when you attempt to compile and run the following code?

```
public class Sandys{
private int court;
public static void main(String argv[]){
        Sandys s = new Sandys(99);
        System.out.println(s.court);
        }
Sandys(int ballcount){
        court=ballcount;
        }
}
```

1) Compile time error, the variable court is defined as private
2) Compile time error, s is not initialized when the System.out method is called
3) Compilation and execution with no output
4) Compilation and run with an output of 99

Answer to Question 45)

Question 46)

Which of the following statements are true?

1) A method cannot be overloaded to be less public in a child class
2) To be overridden a method must have the same name and parameter types
3) To be overridden a method must have the same name, parameter and return types
4) An overridden method must have the same name, parameter names and parameter types

Answer to Question 46)

Question 47)

What will happen when you attempt to compile and run the following code?

```
class Base{
Base(){
        System.out.println("Base");
        }
}
public class Checket extends Base{
public static void main(String argv[]){
        Checket c = new Checket();
        super();
        }
Checket(){
        System.out.println("Checket");
        }
}
```

1) Compile time error
2) Checket followed by Base
3) Base followed by Checket
4) runtime error

Answer to Question 47)

Question 48)

Which of the following statements are true?

1) Static methods cannot be overriden to be non static
2) Static methods cannot be overloaded
3) Private methods cannot be overloaded
4) An overloaded method cannot throw exceptions not checked in the base class

Answer to Question 48)

Question 49)

Which of the following statements are true?

1) The automatic garbage collection of the JVM prevents programs from ever running out of memory
2) A program can suggest that garbage collection be performed but not force it

3) Garbage collection is platform independent
4) An object becomes eligible for garbage collection when all references denoting it are set to null.

Answer to Question 49)

Question 50)

```
Given the following code
public class Sytch{
int x=2000;
public static void main(String argv[]){
        System.out.println("Ms "+argv[1]+"Please pay $"+x);
        }
}
What will happen if you attempt to compile and run this code with the command line
java Sytch Jones Diggle
```

1) Compilation and output of Ms Diggle Please pay $2000
2) Compile time error
3) Compilation and output of Ms Jones Please pay $2000
4) Compilation but runtime error

Answer to Question 50

Question 51)

What will happen when you attempt to compile and run the following code

```
class Base{
protected int i = 99;
}
public class Ab{
private int i=1;
public static void main(String argv[]){
Ab a = new Ab();
a.hallow();
}
        abstract void hallow(){
        System.out.println("Claines "+i);
        }
}
```

1) Compile time error
2) Compilation and output of Claines 99
3) Compilation and output of Claines 1
4) Compilation and not output at runtime

Answer to Question 51)

Question 52)

You have been asked to create a scheduling system for a hotel and catering organsiation.

You have been given the following information and asked to create a set of classes to represent it.

On the catering side of the organsiation they have

Head Chefs
Chefs
Apprentice Chefs

The system needs to store an employeeid, salary and the holiday entitlement

How would you best represent this information in Java

1) Create classes for Head Chef, Chef, Apprentice Chef and store the other values in fields
2) Create an employee class and derive sub classes for Head Chef, Chef, Apprentice Chef abd store the other values in fields.
3) Create and employee class with fields for Job title and fields for the other values.
4) Create classes for all of the items mentioned and create a container class to represent employees

Answer to Question 52)

Question 53)

You need to read in the lines of a large text file containing tens of megabytes of data. Which of the following would be most suitable for reading in such a file

1) new FileInputStream("file.name")
2) new InputStreamReader(new FileInputStream("file.name"))
3) new BufferedReader(new InputStreamReader(new FileInputStream("file.name")));
4) new RandomAccessFile raf=new RandomAccessFile("myfile.txt","+rw");

Answer to Question 53)

Question 54)

What will happen when you attempt to compile and run the following code?

```
public class Inc{
public static void main(String argv[]){
                Inc inc = new Inc();
                int i =0;
                inc.fermin(i);
                i = i++;
                System.out.println(i);
        }
        void fermin(int i){
                i++;
```

```
        }
}
```

1) Compile time error
2) Output of 2
3) Output of 1
4) Output of 0

Answer to Question 55)

Question 55)

What will happen when you attempt to compile and run the following code?

```
public class Agg{
static public long i=10;
public static void main(String argv[]){
        switch(i){
                default:
                System.out.println("no value given");
                case 1:
                System.out.println("one");
                case 10:
                System.out.println("ten");
                case 5:
                System.out.println("five");
        }
    }
}
```

1) Compile time error
2) Output of "ten" followed by "five"
3) Output of "ten"
4) Compilation and run time error because of location of default

Answer to question 55)

Question 56)

Given the following class

```
public class ZeroPrint{
public static void main(String argv[]){
        int i =0;
        //Here
        }
}
```

Which of the following lines if placed after the comment //Here will print out 0.

1) System.out.println(i++);
2) System.out.println(i+' 0');

3) System.out.println(i);
4) System.out.println(i--);

Answer to Question 56)

Question 57)

Answers

Answer to Question 1)

Objective 4.5)

1) float f=1/3;
2) int i=1/3;
4) double d=999d;

The fact that option 3 does not compile may be a surprise. The problem is because the default type for a number with a decimal component is a double and not a float. The additional trailing d in the option with 999 doesn't help, but it doesn't harm.

http://www.software.u-net.com/J2Tutor/04_05Tut.htm

Answer to Question 2)

Objective 4.3)

2) new

The option NULL (note the upper case letter) is definitely not a keyword. There is some discussion as to if null is a keyword but for the purpose of the exam you should probably assume it is a keyword.

The option instanceOf is a bit of a misleading option that would probably not occur on the exam. The real keyword is *instanceof* (note that the of has no capital letter O). I had the incorrect version in an earlier version of this tutorial as it looks more likely to my eyes. The *instanceof* keyword looks like a method, but it is actually an operator.

The option *wend* is probably valid in some other language to indicate the end of a *while* loop, but Java has no such keyword.

http://www.software.u-net.com/J2Tutor/04_03Tut.htm

Answer to Question 3)

Objective 4.5)

1) System.out.println(1+1);
2) int i=2+' 2' ;
Option 3 is not valid because single quotes are used to indicate a character constant and not a string.
Option 4 will not compile because 255 is out of the range of a byte

http://www.software.u-net.com/J2Tutor/04_05Tut.htm

Answer to Question 4)

Objective 7.1)

1) The garbage collection algorithm in Java is vendor implemented

Threading and garbage collection are two of the few areas that are platform dependent. This is one of the
reasons why Java is not suitable for realtime programming. It is not a good idea use it to control your plane or nuclear power station. Once an instance of the Integer class has a value it cannot be changed.

http://www.software.u-net.com/J2Tutor/07_01Tut.htm

Answer to Question 5)

Objective 10.1)

(Not on the official sub objectives but this topic does come up on the exam)

2) The RandomAccessFile class allows you to move directly to any point a file.
4) The characteristics of an instance of the File class such as the directory separator, depend on the current underlying operating system

The File class can be considered to represent information about a file rather than a real file object. You can create a file in the underlying operating system by passing an instance of a file to a stream such as FileOutputStream. The file will be created when you call the close method of the stream.

http://www.software.u-net.com/J2Tutor/10 01Tut.htm

Answer to Question 6)

2) The instanceof operator can be used to determine if a reference is an instance of a particular primitive wrapper class

The instanceof operator can only be used to make a static comparison with a class type. Java1.1 added the isInstance method to the class Class to allow you to dynamically determine a class type. The exam does not test you on isInstance.

http://www.software.u-net.com/J2Tutor/15 01Tut.htm#The instanceof operator

---

Answer to Question 7)

Objective 4.1)

4) Interfaces are the Java approach to addressing the single inheritance model, but require implementing classes to create the functionality of the Interfaces.

An interface may contain variables as well as methods. However any variables are final by default and must be assigned values on creation. A class can only extend one other class (single inheritance) but may implement as many interfaces as you like (or is sensible).

http://www.software.u-net.com/J2Tutor/04_01Tut.htm

Answer to Question 8)

Objective 9.1)

None of these are valid statements. The Math class is final and cannot be extended. The max method takes two parameters, round only takes one parameter and there is no mod parameter. You may get questions in the exam that have no apparently correct answer. If you are absolutely sure this is the case, do not check any of the options.

http://www.software.u-net.com/J2Tutor/09_01Tut.htm

Answer to Question 9)

Objective 7.1)

1) The Runnable interface has only one method run that needs to be created in any class that implements it. The start method is used to actually call and start the run method executing.

http://www.software.u-net.com/J2Tutor/07_01Tut.htm

Answer to Question 10)

Objective 4.5)

1) A byte can represent between -128 to 127

The char type is the only unsigned type in Java and thus cannot represent a negative number.

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/04_05ut.htm

Answer to Question 11)

Objective 1.2)

2) Compilation and no output at runtime

Because the method in Base called Base has a return type it is not a constructor and there for does not get called on creation of an instance of its child class In

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/01_02ut.htm

Answer to Question 12)

Objective 4.2)

4) Compilation and output of hello

This type of question is particularly calculated to catch out C/C++ programmers who might expect parameter zero to be the name of the compiler.

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/04_02ut.htm

Answer to Question 13)

Objective 1.2)

1) If a class has any abstract methods it must be declared abstract itself.
3) The final modifier means that a class cannot be sub-classed
4) transient and volatile are Java modifiers

An *abstract* class may have non *abstract* methods. Any class that descends from an *abstract* class must implement the *abstract* methods of the base class or declare them as *abstract* itself.

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/01_02ut.htm

Answer to Question 14)

Objective 1.2)

2) public static void amethod(){}
4) static native void amethod();

Option 1 is not valid because it has braces and the native modifier means that the method can have no body. This is because the body must be implemented in some other language (often C/C++). Option 3 is not valid because private and protected contradict themselves.

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/01_02ut.htm

Answer to Question 15)

Objective 6.2)

4) Constructors are not inherited

Constructors can be marked public, private or protected. Constructors do not have a return type.

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/06_02ut.htm

Answer to Question 16)

Objective 1.3)

2) Compile time error

An error occurs when the class Severn attempts to call the zero parameter constructor in the class *Base*

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/01_03ut.htm

Answer to Question 17)

Objective 1.2)

1) static methods do not have access to the implicit variable called *this*
3) a static may not be overriden to be non-static

The implicit variable *this* refers to the current instant of a class and thus and by its nature a static method cannot have access to it.

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/01_02ut.htm

Answer to Question 18)

Objective 5.1)

1)

```
char c='  1'  ;
System.out.println(c>>1);
```

4)

```
int i=1;
System.out.println(i<<1);
```

Be aware that Integer (not the upper case I) is a wrapper class and thus cannot be treated like a primitive. The fact that option 1 will compile may be a surprise, but although the *char* type is normally used to store character types, it is actually an unsigned integer type. The reason option 3 does not compile is that Java has a >>> operator but not a <<< operator.

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/05_01Tut.htm

Answer to Question 19)

Objective 4.6)

2) An event listener may be removed from a component
3) The ActionListener interface has no corresponding Adapter class

A component may have multiple event listeners attached. Thus a field may need to respond to both the mouse and the keyboard, requiring multiple event handlers. The ActionListener has not matching Adapter class because it has only one method, the idea of the Adapter classes is to eliminate the need to create blank methods.

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/04_06Tut.htm

Answer to Question 20)

Objective 4.3)

3) transient
4) volatile

Option 1, sizeof is designed to catch out the C/C++ programmers. Java does not have a sizeof keyword as the size of primitives should be consistent on all Java implementations. Although a program needs a main method with the standard signature to start up it is not a keyword. The real keywords are less commonly used and therefore might not be so familiar to you.

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/04_03Tut.htm

Answer to Question 21)

Objective 1.3)

3) The default constructor takes no parameters
4) The default constructor is not created if the class has any constructors of its own.

Option 1 is fairly obviously wrong as constructors never have a return type. Option 2 is very dubious as well as Java does not offer void as a type for a method or constructor.

For more information on this topic go to

http://www.software.u-net.com/J2Tutor/01_03Tut.htm

Answer to Question 22)

Objective 4.1)

1) All of the variables in an interface are implicitly static
2) All of the variables in an interface are implicitly final
3) All of the methods in an interface are implictly abstract

All the variables in an interface are implicitly static and final. Any methods in an interface have no body, so may not access any type of variable

http://www.software.u-net.com/J2Tutor/04_01Tut.htm

Answer to Question 23)

2) The + operator is overloaded for concatenation for the String class

In Java Strings are implemented as a class within the Java.lang package with the special distinction that the + operator is overloaded. If you thought that the String class is implemented as a char array, you may have a head full of C/++ that needs emptying. There is not "wrapper class" for String as wrappers are only for primitive types.

If you are surprised that option 4 is not a correct answer it is because length is a method for the String class, but a property for and array and it is easy to get the two confused.

http://www.software.u-net.com/J2Tutor/05_01Tut.htm

Answer to Question 24)

1) A method in an interface must not have a body
3) A class may extends one other class plus many interfaces

A class accesses an interface using the implements keyword (not uses)

http://www.software.u-net.com/J2Tutor/04_01Tut.htm

Answer to Question 25)

3) The following statement will produce a result of zero, System.out.println(1 >>1);

Although you might not know the exact result of the operation -1 >>> 2 a knowledge of the way the bits will be shifted will tell you that the result is not plus 1. (The result is more like 1073741823) There is no such Java operator as the unsigned left shift. Although it is normally used for storing characters rather than numbers the char Java primitive is actually an unsigned integer type.

http://www.software.u-net.com/J2Tutor/05_01Tut.htm

Answer to Question 26)

2) Arrays are initialized to default values wherever they are created

You can find the size of an array using the length field. The method *length* is used to return the number of characters in a String. An array can contain elements of any type but they must all be of the same type. The size of an array is fixed at creation. If you want to change its size you can of course create a new array and assign the old one to it. A more flexible approach can be to use a collection class such as Vector.

http://www.software.u-net.com/J2Tutor/04_04Tut.htm

Answer to Question 27)

2) Output of "Hello Crowle"

This code is an example of a short circutied operator. Because the first operand of the || (or) operator returns true Java sees no reason to evaluate the second. Whatever the value of the second the overall result will always be true. Thus the method called place is never called.

Answer to Question 28)

4) none of the above;

You may access methods of a direct parent class through the use of super but classes further up the hierarchy are not visible.

Answer to Question 29)

2) A method with the same name completly replaces the functionality of a method earlier in the hierarchy

Option 3 is more like a description of overloading. I like to remind myself of the difference between overloading and overriding in that an overriden method is like something overriden in the road, it is squashed, flat no longer used and replaced by something else. An overloaded method has been given extra work to do (it is loaded up with work), but it is still being used in its original format. This is just my little mind trick and doesn' t match to anything that Java is doing.

Answer to Question 30)

2) The / operator is used to divide one value by another

3) The # symbol may not be used as the first character of a variable

The % is the modulo operator and returns the remainder after a division. Thus 10 % 3=1
The $ symbol may be used as the first character of a variable, but I would suggest that it is generally not a good idea. The # symbol cannot be used anywhere in the name of a variable. Knowing if a variable can start with the # or $ characters may seem like arbitrary and non essential knowlege but questions like this do come up on the exam.

Answer to Question 31)

1) The default layout manager for an Applet is FlowLayout
4) The FlowLayout manager attempts to honor the preferred size of any components

The default layout manager fror an Application is BorderLayout. An applet will use the default of FlowLayout if one is not specifically applied.

Answer to Question 32)

3) Only one instance of a static variable will exist for any amount of class instances

Option 1) is more a description of a final variable. Option 2 is designed to fool Visual Basic programmers like me as this is how you can use the keyword static in VB. The modifier static can be applied to a class, method or variable.

Answer to Question 33)

1) Java uses a system called UTF for I/O to support international character sets
3) An instance of FileInputStream may not be chained to an instance of FileOutputStream
4) File I/O activities requires use of Exception handling

Internally Java uses Unicode which are 16 bit characters. For I/O Java uses UTF which may be more thatn 16 bits per character.
Generally InputStreams can only be chained to other OutputStreams and OutputStreams can only be chained to other OutputStreams. The piped streams are an exception to this.

Answer to Question 34)

1) Compile time error

It wil produce an error like "Abstract and native method can' t have a body. This is typical of the more misleading question where you might think it is asking you about the circumstances under which the finally clause runs, but actually it is about something else.

Answer to Question 35)

2) Compilation and run with the output "Running"

This is perfectly legitimate if useless sample of creating an instnace of a Thread and causing its run method to execute via a call to the start method. The Thread class is part of the core java.lang package and does not need any explicit import statement. The reference to a Thread target is an attempt to mislead with a reference to the method of using the Runnable interface instead of simply inheriting from the Thread super class.

Answer to Question 36)

1) RandomAccessFile raf=new RandomAccessFile("myfile.txt","+rw");

The RandomAccessFile is an anomoly in the Java I/O architecture. It descends directly from Object and is not part of the Streams architecture.

Answer to Question 37)

2) public int amethod(int i, int j) {return 99;}
3) protected void amethod (long l){}
4) private void anothermethod(){}

Option 1 will not compile on two counts. One is the obvious one that it claims to return an integer. The other is that it is effectivly an attempt to redefine a method within the same class. The change of name of the parameter from i to z has no effect and a method cannot be overriden within the same class.

Answer to Question 38)

1) true

That statemetn is a direct quote from the Java Language Specification for JDK 1.1 as published by O' reilly. This is sometimes known as the JLS. I was looking up this topic as it came up in discussion as to if a top level nested class should be considered an inner class. I believe it should be for the purpose of the exam.

Answer to Question 39)

4) The code will compile with

There are no restrictions on the level of nesting for inner/nested classes. Inner classes may be marked private. The main method is not declared as public static void main, and assuming that the

commandline was java Droitwich it would not be invoked anyway.

Answer to Question 40)

1) super.oak=1;
2) oak=33;
3) Base.oak=22;

Because the variable oak is declared as static only one copy of it will exist. Thus it can be changed either through the name of its class or through the name of any instance of that class. Because it is created as an integer it canot be assigned a fractional component without a cast.

Answer to Question 41)

4) Use the getText field of a Textfield and use the parseInt method of the Integer class

Here is an example of how you might do this

Integer.parseInt(txtInputValue.getText());

I' m not sure that a question on this actually will come up in the exam but it is a very useful thing to know in the real world.

Objective 4.6)

Answer to Question 42)

4) none of the above

The wrapper classes are immutable. Once the value has been set it cannot be changed. A common use of the wrapper classes is to take advantage of their static methods such as Integer.parseInt(String s) that will returns an integer if the String contains one.

Answer to Question 43)

2) constructors cannot be overriden
4) Constructor invokation occurs from the current class up the hierarchy to the ancestor class

Overloading constructors is a key technique to allow multiple ways of initialising classes. By definition constructors have no return values so option 3 makes no sense.

Answer to Question 44)

yield is a static method inherited from object and causes whatever thread is currently executing to yield its cycles.

2) yeild()

Answer to Question 45)

4) Compilation and run with an output of 99

The fact that the variable court is declared as private does not stop the constructor from being able to initialise it.

Answer to Question 46)

3) To be overriden a method must have the same name, parameter and return types

Option 1 is a sneaky one in that it should read overriden not overloaded. An overriden method must also have the same return type. Parameter names are purely a programmer convenience and are not a factor in either overloading and overriding. Parameter order is a factor however.

Answer to Question 47)

1) Compile time error

With the sun JDK it will produce the following error

"Only constructors can invoke constructors".

If you took out the call to super that causes this error the program would compile and at runtime it would output Base and then Checket as constructors are called from the oldest ancestor class downwards.

Answer to Question 48)

1) Static methods cannot be overriden to be non static

The JDK1.1 compiler will issue an error message "static methods cannot be overriden" if you atempt to do this. There is no logic or reason why private methods should not be overloaded. Option 4 is a jumbled up version of the limitations of exceptions for overriden methods

---

Answer to Question 49)

2) A program can suggest that garbage collection be performed but not force it
4) A reference becomes eligable for garbage collection when it is assigned to null

If a program keeps creating new references without any being discarded it may run out of memory. Unlike most aspects of Java garbage collection is platform dependent.

Answer to Question 50)

2) Compile time error

The main method is static and cannot access the non static variable x

---

Answer to Question 51)
Objective 1.2)

1) Compile time error

When compiled with JDK 1.1 the following error is produced.

Abstract and native methods can' t have a body: void hallow() abstract void hallow()

http://www.software.u-net.com/J2Tutor/01_02Tut.htm

Answer to Question 52)
Objective 6.1)

3) Create and employee class with fields for Job title and fields for the other values.

These questions can appear tricky as the whole business of designing class structures is more art than science. It is asking you to decide if an item of data is best represented by the "Is a" or "Has a" relationship. Thus in this case any of the job titles mentioned will always refer to something that "Is a" employee. However the employee "has a" job title that might change.

One of the important points is to ask yourself when creating a class "Could this change into another class at some point in the future". Thus in this example an apprentice chef would hope one day to turn into a chef and if she is very good will one day be head chef. Few other mock exams seem to have this type of questions but they di come up in the real exam.

Answer to Question 53)

3) new BufferedReader(new InputStreamReader(new FileInputStream("file.name")));

The key to this question is that it asks about tens of megabytes of data, implying that performance is

an issue. A Buffered Reader will optimise the performance of accessing a file. Although the objectives do not specifically mention it questions on I/O do come up on the exam.

Answer to Question 54)

4) Output of 0

The method fermin only receives a copy of the variable i and any modifications to it are not reflected in the version in the calling method. The post increment operator ++ effectivly modifes the value of i after the initial value has been assiged to the left hand side of the equals operator. This can be a very tricky conept to understand

---

Answer to Question 55)
Objective 2.2)

1) Compile time error

This might be considered a "gocha" or deliberate attempt to mislead you because i has been given the data type of long and the parameter must be either a byte, char, short or int. If you attempt to compile this code with JDK 1.2 you will get an error that says something like "Incompatible type for switch, Explicit cast needed to convert long to int. Answering with option 2 would have been reasonable because if the parameter had been an integer type the lack of break statements would have caused this output. If you gave either of the answers you should probably revise the subject.

Answer to Question 56)

1) System.out.println(i++);
3) System.out.println(i);
4) System.out.println(i--);

The options for this question might look suspiciously easy if you are not aware of the effects of the post-increment operators. The ++ and -- operations for examples 1 and 4 only come into effect after the output operations, ie after whatever else is done to them on that line of code. Option 2 should be fairly obvious as you should know that the single quote characters indicate a char value, ie storing the character rather than the numberical value for 0.

Answer to Question 57)

End of document