Java Programmer Certification Mock Exam No 2

Last Updated 29 February 2000

60 Questions

This is my mock Exam No 2 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, but please read this document first
http://www.software.u-net.com/getanswers.htm

If you obtained this exam from some other source please email me to let me know and check out the latest version at http://www.marcusgreen.co.uk.

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 untill you can see the answer to Question 60 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)

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

```
abstract class Base{
        abstract public void myfunc();
        public void another(){
        System.out.println("Another method");
        }
}
public class Abs extends Base{
        public static void main(String argv[]){
        Abs a = new Abs();
        a.amethod();
        }
        public void myfunc(){
                System.out.println("My func");
                }
        public void amethod(){
        myfunc();

        }
}
```

1) The code will compile and run, printing out the words "My Func"
2) The compiler will complain that the Base class has non abstract methods
3) The code will compile but complain at run time that the Base class has non abstract methods
4) The compiler will complain that the method myfunc in the base class has no body, nobody at all to looove it

Answer to Question 1

Question 2)

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

```
public class MyMain{
public static void main(String argv){
        System.out.println("Hello cruel world");
        }
}
```

1) The compiler will complain that main is a reserved word and cannot be used for a class
2) The code will compile and when run will print out "Hello cruel world"
3) The code will compile but will complain at run time that no constructor is defined
4) The code will compile but will complain at run time that main is not correctly defined

Answer to Question 2

Question 3)

Which of the following are Java modifiers?

1) public
2) private
3) friendly

4) transient
4) vagrant

Answer to Question 3

Question 4)

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

```
class Base{
        abstract public void myfunc();
        public void another(){
        System.out.println("Another method");
        }
}
public class Abs extends Base{
        public static void main(String argv[]){
        Abs a = new Abs();
        a.amethod();
        }
        public void myfunc(){
                System.out.println("My func");
                }
        public void amethod(){
        myfunc();

        }
}
```

1) The code will compile and run, printing out the words "My Func"
2) The compiler will complain that the Base class is not declared as abstract.
3) The code will compile but complain at run time that the Base class has non abstract methods
4) The compiler will complain that the method myfunc in the base class has no body, nobody at all to looove it

Answer to Question 4

Question 5)

Why might you define a method as native?

1) To get to access hardware that Java does not know about
2) To define a new data type such as an unsigned integer
3) To write optimised code for performance in a language such as C/C++
4) To overcome the limitation of the private scope of a method

Answer to Question 5

Question 6)

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

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

1) Compile time error indicating that a class with any final methods must be declared final itself
2) Compile time error indicating that you cannot inherit from a class with final methods
3) Run time error indicating that Base is not defined as final
4) Success in compilation and output of "amethod" at run time.

Answer to Question 6

---

7) What will happen when you attempt to compile and run this code?

```
public class Mod{
public static void main(String argv[]){
    }
        public static native void amethod();
}
```

1) Error at compilation: native method cannot be static
2) Error at compilation native method must return value
3) Compilation but error at run time unless you have made code containing native amethod available
4) Compilation and execution without error

Answer to Question 7

---

8) What will happen when you attempt to compile and run this code?

```
private class Base{}
public class Vis{
        transient int  iVal;
        public static void main(String elephant[]){
        }
}
```

1)Compile time error: Base cannot be private
2)Compile time error indicating that an integer cannot be transient
3)Compile time error transient not a data type
4)Compile time error malformed main method

Answer to Question 8

---

9) What happens when you attempt to compile and run these two files in the same directory?

```
//File P1.java

package MyPackage;
class P1{
void afancymethod(){
        System.out.println("What a fancy method");
        }
}

//File P2.java

public class P2 extends P1{
afancymethod();
}
```

1) Both compile and P2 outputs "What a fancy method" when run
2) Neither will compile
3) Both compile but P2 has an error at run time
4) P1 compiles cleanly but P2 has an error at compile time

Answer to Question 9

---

10) You want to find out the value of the last element of an array. You write the following code. What will happen when you compile and run it.?

```
public class MyAr{
public static void main(String argv[]){
        int[] i = new int[5];
        System.out.println(i[5]);
        }
}
```

1) An error at compile time
2) An error at run time
3) The value 0 will be output
4) The string "null" will be output

Answer to Question 10

---

11) You want to loop through an array and stop when you come to the last element. Being a good java programmer and forgetting everything you ever knew about C/C++ you know that arrays contain information about their size. Which of the following can you use?

1)myarray.length();

2)myarray.length;
3)myarray.size
4)myarray.size();

Answer to Question 11

Question 12)

What best describes the appearance of an application with the following code?

```
import java.awt.*;

public class FlowAp extends Frame{

public static void main(String argv[]){
    FlowAp fa=new FlowAp();
    fa.setSize(400,300);
    fa.setVisible(true);
}
FlowAp(){
        add(new Button("One"));
        add(new Button("Two"));
        add(new Button("Three"));
        add(new Button("Four"));
    }//End of constructor

}//End of Application
```

1) A Frame with buttons marked One to Four placed on each edge.
2) A Frame with buutons marked One to four running from the top to bottom
3) A Frame with one large button marked Four in the Centre
4) An Error at run time indicating you have not set a LayoutManager

Answer to Question 12

Question 13)

How do you indicate where a component will be positioned using Flowlayout?

1) North, South,East,West
2) Assign a row/column grid reference
3) Pass a X/Y percentage parameter to the add method
4) Do nothing, the FlowLayout will position the component

Answer to Question 13)

Question 14)

How do you change the current layout manager for a container

1) Use the setLayout method
2) Once created you cannot change the current layout manager of a component
3) Use the setLayoutManager method
4) Use the updateLayout method

Answer to Question 14)

Question 15)

Which of the following are fields of the GridBagConstraints class?

1) ipadx
2) fill
3) insets
4) width

Answer to Question 15)

Question 16)

What most closely matches the appearance when this code runs?

```
import java.awt.*;

public class CompLay extends Frame{
public static void main(String argv[]){
    CompLay cl = new CompLay();
    }

CompLay(){
    Panel p = new Panel();
    p.setBackground(Color.pink);
    p.add(new Button("One"));
    p.add(new Button("Two"));
    p.add(new Button("Three"));
    add("South",p);
    setLayout(new FlowLayout());
    setSize(300,300);
    setVisible(true);
    }
}
```

1) The buttons will run from left to right along the bottom of the Frame
2) The buttons will run from left to right along the top of the frame
3) The buttons will not be displayed
4) Only button three will show occupying all of the frame

Answer to Question 16)

Question 17)

Which statements are correct about the *anchor* field?

1) It is a field of the GridBagLayout manager for controlling component placement
2) It is a field of the GridBagConstraints class for controlling component placement
3) A valid setting for the anchor field is GridBagConstraints.NORTH
4) The anchor field controls the height of components added to a container

Answer to Question 17)

Question 18)

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

```
public class Bground extends Thread{

public static void main(String argv[]){
                Bground b = new Bground();
                b.run();
        }
        public void start(){
                for (int i = 0; i <10; i++){
                        System.out.println("Value of i = " + i);
                }
        }

}
```

1) A compile time error indicating that no run method is defined for the Thread class
2) A run time error indicating that no run method is defined for the Thread class
3) Clean compile and at run time the values 0 to 9 are printed out
4) Clean compile but no output at runtime

Answer to Question 18)

Question 19)

10)When using the GridBagLayout manager, each new component requires a new instance of the GridBagConstraints class. Is this statement

1) true
2) false

Answer to Question 19)

Question 20)

Which most closely matches a description of a Java Map?

1) A vector of arrays for a 2D geographic representation
2) A class for containing unique array elements
3) A class for containing unique vector elements
4) An interface that ensures that implementing classes cannot contain duplicate keys

Answer to Question 20)

Question 21)

How does the set collection deal with duplicate elements?

1) An exception is thrown if you attempt to add an element with a duplicate value
2) The *add* method returns false if you attempt to add an element with a duplicate value
3) A set may contain elements that return duplicate values from a call to the equals method
4) Duplicate values will cause an error at compile time

Answer to Question 21)

Question 22)

What can cause a thread to stop executing?

1) The program exits via a call to System.exit(0);
2) Another thread is given a higher priority
3) A call to the thread's stop method.
4) A call to the halt method of the Thread class

Answer to Question 22)

Question 23)

For a class defined inside a method, what rule governs access to the variables of the enclosing method?

1) The class can access any variable
2) The class can only access static variables
3) The class can only access transient variables
4) The class can only access final variables

Answer to Question 23)

Question 24)

Under what circumstances might you use the yield method of the Thread class

1) To call from the currently running thread to allow another thread of the same priority to run
2) To call on a waiting thread to allow it to run
3) To allow a thread of higher priority to run
4) To call from the currently running thread with a parameter designating which thread should be allowed to run

Answer to Question 24)

Question 25)

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

```
public class Hope{
public static void main(String argv[]){
 Hope h = new Hope();
 }
protected Hope(){
 for(int i =0; i <10; i ++){
  System.out.println(i);
  }
 }
}
```

1) Compilation error: Constructors cannot be declared protected
2) Run time error: Constructors cannot be declared protected
3) Compilation and running with output 0 to 10
4) Compilation and running with output 0 to 9

Answer to Question 25)

Question 26)

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

```
public class MySwitch{

public static void main(String argv[]){
    MySwitch ms= new MySwitch();
    ms.amethod();
    }

public void amethod(){

    int k=10;
        switch(k){
        default: //Put the default at the bottom, not here
            System.out.println("This is the default output");
            break;
         case 10:
            System.out.println("ten");
         case 20:
```

```
            System.out.println("twenty");
         break;
        }
    }
}
```

1) None of these options
2) Compile time errror target of switch must be an integral type
3) Compile and run with output "This is the default output"
4) Compile and run with output "ten"

Answer to Question 26)

Question 27)

Which of the following is the correct syntax for suggesting that the JVM performs garbage collection

1) System.free();
2) System.setGarbageCollection();
3) System.out.gc();
4) System.gc();

Answer to Question 27)

Question 28)

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

```
public class As{
int i = 10;
int j;
char z= 1;
boolean b;
public static void main(String argv[]){
                As a = new As();
                a.amethod();
         }
         public void amethod(){
         System.out.println(j);
         System.out.println(b);
         }
}
```

1) Compilation succeeds and at run time an output of 0 and false
2) Compilation succeeds and at run time an output of 0 and true
3) Compile time error b is not initialised
4) Compile time error z must be assigned a char value

Answer to Question 28)

Question 29)

What will happen when you attempt to compile and run the following code with the command line "hello there"

```
public class Arg{
String[] MyArg;
        public static void main(String argv[]){
        MyArg=argv;
        }
        public void amethod(){
                System.out.println(argv[1]);
        }
}
```

1) Compile time error
2) Compilation and output of "hello"
3) Compilation and output of "there"
4) None of the above

Answer to Question 29)

Question 30)

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

```
public class StrEq{

public static void main(String argv[]){
        StrEq s = new StrEq();
        }
        private StrEq(){
                String s = "Marcus";
                String s2 = new String("Marcus");
                if(s == s2){
                        System.out.println("we have a match");
                        }else{
                        System.out.println("Not equal");
                }
       }
}
```

1) Compile time error caused by private constructor
2) Output of "we have a match"
3) Output of "Not equal"
4) Compile time error by attempting to compare strings using ==

Answer to Question 30)

Question 31)

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

```
import java.io.*;

class Base{
public static void amethod()throws FileNotFoundException{}
}

public class ExcepDemo extends Base{
public static void main(String argv[]){
       ExcepDemo e = new ExcepDemo();
}
public static void amethod(){}

protected ExcepDemo(){
 try{
  DataInputStream din = new DataInputStream(System.in);
  System.out.println("Pausing");
  din.readChar();
  System.out.println("Continuing");
  this.amethod();
  }catch(IOException ioe) {}
}

}
```

1)Compile time error caused by protected constructor
2) Compile time error caused by *amethod* not declaring Exception
3) Runtime error caused by amethod not declaring Exception
4) Compile and run with output of "Pausing" and "Continuing" after a key is hit

Answer to Question 31)

Question 32)

What will happen when you attempt to compile and run this program

```
public class Outer{
public String name = "Outer";
public static void main(String argv[]){
        Inner i = new Inner();
        i.showName();
    }//End of main

        private class Inner{
        String name =new String("Inner");
                void showName(){
                        System.out.println(name);
                }
        }//End of Inner class

}
```

1) Compile and run with output of "Outer"
2) Compile and run with output of "Inner"
3) Compile time error because Inner is declared as private

4) Compile time error because of the line creating the instance of Inner

Answer to Question to 32

Question 33)

What will happen when you attempt to compile and run this code

```
//Demonstration of event handling
import java.awt.event.*;
import java.awt.*;

public class MyWc extends Frame implements WindowListener{
public static void main(String argv[]){
        MyWc mwc = new MyWc();
        }
        public void windowClosing(WindowEvent we){
                System.exit(0);
                }//End of windowClosing

      public void  MyWc(){
        setSize(300,300);
        setVisible(true);
        }
}//End of class
```

1) Error at compile time
2) Visible Frame created that that can be closed
3) Compilation but no output at run time
4) Error at compile time because of comment before *import* statements

Answer to Question 33)

Question 34)

Which option most fully describes will happen when you attempt to compile and run the following code

```
public class MyAr{
public static void main(String argv[]) {
        MyAr m = new MyAr();
        m.amethod();
        }
        public void amethod(){
        static int i;
        System.out.println(i);
        }
}
```

1) Compilation and output of the value 0
2) Compile time error because i has not been initialized
3) Compilation and output of null
4) Compile time error

Answer to Question 34)

Question 35)

Which of the following will compile correctly

1) short myshort = 99S;
2) String name = 'Excellent tutorial Mr Green';
3) char c = 17c;
4)int z = 015;

Answer to Question 35)

Question 36)

Which of the following are Java key words
1)double
2)Switch
3)then
4)instanceof

Answer to Question 36)

Question 37
What will be output by the following line?

```
System.out.println(Math.floor(-2.1));
```

1) -2
2) 2.0
3) -3
4) -3.0

Answer to Question 37)

Question 38)

Given the following main method in a class called Cycle and a command line of

```
java Cycle one two
```

what will be output?

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

1) None of these options

2) cycle
3) one
4) two

Answer to Question 38)

Question 39)

Which of the following statements are true?

1) At the root of the collection hierarchy is a class called Collection
2) The collection interface contains a method called enumerator
3) The interator method returns an instance of the Vector class
4) The set interface is designed for unique elements

Answer to Question 39)

Question 40)

Which of the following statements are correct?

1) If multiple listeners are added to a component only events for the last listener added will be processed
2) If multiple listeners are added to a component the events will be processed for all but with no guarantee in the order
3) Adding multiple listeners to a comnponent will cause a compile time error
4) You may remove as well add listeners to a component.

Answer to Question 40)

Question 41)

Given the following code

```
class Base{}

public class MyCast extends Base{
static boolean b1=false;
static int i = -1;
static double d = 10.1;

public static void main(String argv[]){
        MyCast m = new MyCast();
        Base b = new Base();
        //Here
        }
}
```

Which of the following, if inserted at the comment //Here will allow the code to compile and run without error

1) b=m;
2) m=b;
3) d =i;
4) b1 =i;

Answer to Question 41)

Question 42)

Which of the following statements about threading are true

1) You can only obtain a mutually exclusive lock on methods in a class that extends Thread or implements runnable
2) You can obtain a mutually exclusive lock on any object
3) A thread can obtain a mutually exclusive lock on a method declared with the keyword synchronized
4) Thread scheduling algorithms are platform dependent

Answer to Question 42)

Question 43)

Your chief Software designer has shown you a sketch of the new Computer parts system she is about to create. At the top of the hierarchy is a Class called Computer and under this are two child classes. One is called LinuxPC and one is called WindowsPC. The main difference between the two is that one runs the Linux operating System and the other runs the Windows System (of course another difference is that one needs constant re-booting and the other runs reliably). Under the WindowsPC are two Sub classes one called Server and one Called Workstation. How might you appraise your designers work?

1) Give the goahead for further design using the current scheme
2) Ask for a re-design of the hierarchy with changing the Operating System to a field rather than Class type
3) Ask for the option of WindowsPC to be removed as it will soon be obsolete
4) Change the hierarchy to remove the need for the superfluous Computer Class.

Answer to Question 43)

Question 44)

Objective 4.1)

Which of the following statements are true

1) An inner class may be defined as static

2) There are NO circumstances where an inner class may be defined as private
3) An anonymous class may have only one constructor
4) An inner class may extend another class

Answer to Question 44)

Question 45)

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

```
int Output=10;
boolean b1 = false;
if((b1==true) && ((Output+=10)==20)){
    System.out.println("We are equal "+Output);
    }else
    {
    System.out.println("Not equal! "+Output);
}
```

1) Compile error, attempting to peform binary comparison on logical data type
2) Compilation and output of "We are equal 10"
3) Compilation and output of "Not equal! 20"
4) Compilation and output of "Not equal! 10"

Answer to Question 45)

Question 46)

Given the following variables which of the following lines will compile without error?

```
String s = "Hello";
long l = 99;
double d = 1.11;
int i = 1;
int j = 0;

1) j= i <<s;
2) j= i<<j;
3) j=i<<d;
4)j=i<<l;
```

Answer to Question 46)

Question 47)

What will be output by the following line of code?

System.out.println(010|4);

1) 14
2) 0
3) 6

4) 12

Answer to Question 47)

Question 48)

Given the following variables

```
char c = 'c';
int i = 10;
double d = 10;
long l = 1;
String s = "Hello";
```

Which of the following will compile without error?

1)c=c+i;
2)s+=i;
3)i+=s;
4)c+=s;

Answer to Question 48)

Question 49)

Which of the following will compile without error?

1) File f = new File("/","autoexec.bat");
2) DataInputStream d = new DataInputStream(System.in);
3) OutputStreamWriter o = new OutputStreamWriter(System.out);
4) RandomAccessFile r = new RandomAccessFile("OutFile");

Answer to Question 49)

Question 50)

Given the folowing classes which of the following will compile without error?

```
interface IFace{}
class CFace implements IFace{}
class Base{}

public class ObRef extends Base{
public static void main(String argv[]){
        ObRef ob = new ObRef();
        Base b = new Base();
        Object o1 = new Object();
        IFace o2 = new CFace();
        }
}
```

```
1)o1=o2;
2)b=ob;
3)ob=b;
4)o1=b;
```

Answer to Question 50)

Question 51)

Given the following code what will be the output?

```
class ValHold{
        public int i = 10;
}
public class ObParm{
public static void main(String argv[]){
        ObParm o = new ObParm();
        o.amethod();
        }
        public void amethod(){
                int i = 99;
                ValHold v = new ValHold();
                v.i=30;
                another(v,i);
                System.out.println(v.i);

        }//End of amethod

        public void another(ValHold v, int i){
                i=0;
                v.i = 20;
                ValHold vh = new ValHold();
                v =  vh;
                System.out.println(v.i+ " "+i);
        }//End of another
}
```

1) 10,0, 30
2) 20,0,30
3) 20,99,30
4) 10,0,20

Answer to Question 51)

Question 52)

Given the following class definition, which of the following methods could be legally placed after the comment

```
//Here

public class Rid{
        public void amethod(int i, String s){}
```

```
        //Here
}
```

1)public void amethod(String s, int i){}
2)public int amethod(int i, String s){}
3)public void amethod(int i, String mystring){}
4) public void Amethod(int i, String s) {}

Answer to Question 52)

Question 53)

Given the following class definition which of the following can be legally placed after the comment line
//Here ?

```
class Base{
public Base(int i){}
}

public class MyOver extends Base{
public static void main(String arg[]){
                MyOver m = new MyOver(10);
                }
        MyOver(int i){
                super(i);
        }
        MyOver(String s, int i){
                this(i);
                                //Here

        }
}
```

1)MyOver m = new MyOver();
2)super();
3)this("Hello",10);
4)Base b = new Base(10);

Answer to Question 53)

Question 54)

Given the following class definition, which of the following statements would be legal after the comment //Here

```
class InOut{
String s= new String("Between");


        public void amethod(final int iArgs){
        int iam;
               class Bicycle{
                        public void sayHello(){
```

```
                                                    //Here
                        }//End of bicycle class
                }
         }//End of amethod
        public void another(){
        int iOther;
        }
}
```

1)System.out.println(s);
2) System.out.println(iOther);
3) System.out.println(iam);
4) System.out.println(iArgs);

Answer to Question 54)

Question 55)

Which of the following are methods of the Thread class?

1) yield()
2) sleep(long msec)
3) go()
4) stop()

Answer to Question 55)

Question 56)

Which of the following methods are members of the Vector class and allow you to input a new element

1) addElement
2) insert
3) append
4) addItem

Answer to Question 56)

Question 57)

Which of the following statements are true?

1) Adding more classes via import statements will cause a performance overhead, only import classes you actually use.
2) Under no circumstances can a class be defined with the *private* modifier

3) A inner class may under some circumstances be defined with the *protected* modifier
4) An interface cannot be instantiated

Answer 57)

Question 58)

Which of the following are correct event handling methods

1) mousePressed(MouseEvent e){}
2) MousePressed(MouseClick e){}
3) functionKey(KeyPress k){}
4) componentAdded(ContainerEvent e){}

Answer 58)

Question 59)

Which of the following are methods of the Collection interface?
1) iterator
2) isEmpty
3) toArray
4) setText

Answer 59)

Question 60)

Which of the following best describes the use of the synhronized keyword?

1) Allows two process to run in paralell but to communicate with each other
2) Ensures only one thread at a time may access a class or object
3) Ensures that two or more processes will start and end at the same time
4) Ensures that two or more Threads will start and end at the same time

Answer 60)

Answers

Answer 1)

Objective 1.2)

1) The code will compile and run, printing out the words "My Func"

A class that contains an abstract method must be declared abstract itself, but may contain non abstract methods.

Answer 2)

Objective 4.1)

4) The code will compile but will complain at run time that main is not correctly defined

In this example the parameter is a string not a string array as needed for the correct main method

---

Answer 3)

Objective 4.3)

1) public
2) private
4) transient

The keyword transient is easy to forget as is not frequently used. Although a method may be considered to be friendly like in C++ it is not a Java keyword.

Answer 4)

Objective 1.2)

2) The compiler will complain that the Base class is not declared as abstract.

If a class contains abstract methods it must itself be declared as abstract

---

Answer 5)

Objective 1.2)

1) To get to access hardware that Java does not know about
3) To write optimised code for performance in a language such as C/C++

Answer 6)

Objective 1.2)

4) Success in compilation and output of "amethod" at run time.

A final method cannot be ovverriden in a sub class, but apart from that it does not cause any other restrictions.

Answer 7)

Objective 1.2)

4) Compilation and execution without error

It would cause a run time error if you had a call to amethod though.

Answer 8)

Objective 1.2)

1)Compile time error: Base cannot be private

A top leve (non nested) class cannot be private.

Answer 9)

Objective 1.2)

4) P1 compiles cleanly but P2 has an error at compile time

The package statement in P1.java is the equivalent of placing the file in a different directory to the file P2.java and thus when the compiler tries to compile P2 an error occurs indicating that superclass P1 cannot be found.

Answer 10)

Objective 1.1)

2) An error at run time

This code will compile, but at run-time you will get an ArrayIndexOutOfBounds exception. This becuase counting in Java starts from 0 and so the 5th element of this array would be i[4].

Remember that arrays will always be initialized to default values wherever they are created.

Answer 11)

Objective 1.1)

2)myarray.length;

The String class has a length() method to return the number of characters. I have sometimes become confused between the two.

Answer 12)

Objective 8.2)

3) A Frame with one large button marked Four in the Centre

The default layout manager for a Frame is the BorderLayout manager. This Layout manager defaults to placing components in the centre if no constraint is passed with the call to the add method.

Answer 13)

Objective 8.2)

4) Do nothing, the FlowLayout will position the component

Answer 14)

Objective 8.2)

1) Use the setLayout method

Answer 15)

Objective 8.2)

1) ipadx
2) fill
3) insets

Answer 16)

Objective 8.2)

2) The buttons will run from left to right along the top of the frame

The call to the setLayout(new FlowLayout()) resets the Layout manager for the entire frame and so the buttons end up at the top rather than the bottom.

Answer 17)

Objective 8.2)

2) It is a field of the GridBagConstraints class for controlling component placement
3) A valid settting for the anchor field is GridBagconstraints.NORTH

Answer 18)

Objective 7.1)

4) Clean compile but no output at runtime

This is a bit of a sneaky one as I have swapped around the names of the methods you need to define and call when running a thread. If the for loop were defined in a method called

```
public void run()
```

```
and the call in the main method had been to b.start()
```

The list of values from 0 to 9 would have been output.

Answer 19)

Objective 8.2)

2) false

You can re-use the same instance of the GridBagConstraints when added successive components.

Answer 20)

Objective 10.1)

4) An interface that ensures that implementing classes cannot contain duplicates

---

Answer 21)

Objective 10.1)

2) The *add* method returns false if you attempt to add an element with a duplicate value

I find it a surprise that you do not get an exception.

Answer 22)

Objective 7.1)

1) The program exits via a call to exit(0);
2) The priority of another thread is increased
3) A call to the stop method of the Thread class

Java threads are somewhat platform dependent and you should be carefull when making assumptions about Thread priorities. On some platforms you may find that a Thread with higher priorities gets to "hog" the processor.

Answer 23)

Objective 4.1)

4) The class can only access final variables

Answer 24)

Objective 7.1)

1) To call from the currently running thread to allow another thread of the same priority to run

---

Answer 25)

Objective 6.2)

4) Compilation and running with output 0 to 9

Answer 26)

Objective 2.1)

1) None of these options

Because of the lack of a break statement after the break 10; statement the actual output will be

"ten" followed by "twenty"

Answer 27)

Objective 3.1)

4) System.gc();

Answer 28)

Objective 4.4)

1) Compilation succeeds and at run time an output of 0 and false
The default value for a boolean declared at class level is false, and integer is 0;

Answer 29)

Objective 1.2)

1) Compile time error

You will get an error saying something like "Cant make a static reference to a non static variable". Note that the main method is static.

Answer 30)

Objective 5.2)

3) Output of "Not equal"

Despite the actual character strings matching, using the == operator will simply compare memory location. Because the one string was created with the new operator it will be in a different location in memory to the other string.

Answer 31)

Objective 2.3)

4) Compile and run with output of "Pausing" and "Continuing" after a key is hit

An overriden method in a sub class must not throw Exceptions not thrown in the base class. In the case of the method amethod it throws no exceptions and will thus compile without complain. There is no reason that a constructor cannot be protected.

Answer 32)

Objective 6.3)

4) Compile time error because of the line creating the instance of Inner

This looks like a question about inner classes but it is also a reference to the fact that the main method is static and thus you cannot directly access a non static method. The line causing the error could be fixed by changing it to say

```
Inner i = new Outer().new Inner();
```

Then the code would compile and run producing the output "Inner"

Answer 33)

Objective 4.6)

1) Error at compile time

If you implement an interface you must create bodies for all methods in that interface. This code will produce an error saying that MyWc must be declared abstract because it does not define all of the methods in WindowListener. Option 4 is nonsense as comments can appear anywhere. Option 3 suggesting that it might compile but not produce output is ment to mislead on the basis that what looks like a constructor is actually an ordinary method as it has a return type.

Answer 34)

Objective 1.2)

4) Compile time error

An error will be caused by attempting to define an integer as static within a method. The lifetime of a field within a method is the duration of the running of the method. A static field exists once only for the class. An approach like this does work with Visual Basic.

Answer 35)

Objective 9.5)

4)int z = 015;

The letters c and s do not exist as literal indicators and a String must be enclosed with double quotes, not single as in this case.

Answer 36)

Objective 4.3)

1)double
4)instanceof

Note the upper case S on switch means it is not a keyword and the word then is part of Visual Basic but not Java. Also, *instanceof* looks like a method but is actually a keyword,

---

Answer 37)

Objective 9.2)

4) -3.0

Answer 38)

Objective 4.2)

3) one

Command line parameters start from 0 and fromt he first parameter after the name of the compile (normally Java)

Answer 39)

Objective 10.1)

4) The set is designed for unique elements.

Collection is an interface, not a class. The Collection interface includes a method called iterator. This returns an instance of the Iterator class which has some similarities with Enumerators.
The name set should give away the purpose of the Set interface as it is analogous to the Set concept in relational databases which implies uniquness.

Answer 40)

Objective 8.1)

2) If multiple listeners are added to a component the events will be processed for all but with no

guarantee in the order
4) You may remove as well add listeners to a component.

It ought to be fairly intuitive that a component ought to be able to have multiple listeners. After all, a text field might want to respond to both the mouse and keyboard

---

Answer 41)

Objective 5.1)

1) b=m;
3) d =i;

You can assign up the inheritance tree from a child to a parent but not the other way without an explicit casting. A boolean can only ever be assigned a boolean value.

Answer 42)

Objective 7.3)

2) You can obtain a mutually exclusive lock on any object
3) A thread can obtain a mutually exclusive lock on a method declared with the keyword synchronized
4) Thread scheduling algorithms are platform dependent

Yes that says dependent and not independent.

Answer 43)

Objective 6.1)

2) Ask for a re-design of the hierarchy with changing the Operating System to a field rather than Class type

Of course there are as many ways to design an object hierarchy as ways to pronounce Bjarne Strousjoup, but this is the sort of answer that Sun will proabably be looking for in the exam.

Answer 44)

Objective 4.1)

1) An inner class may be defined as static
4) An inner class may extend another class

A static inner class is also sometimes known as a top level nested class. There is some debate if such

a class should be called an inner class. I tend to think it should be on the basis that it is created inside the opening braces of another class. How could an anonymous class have a constructor?. Remember a constructor is a method with no return type and the same name as the class. Inner classes may be defined as private

---

Answer 45)

Objective 5.3)

4) Compilation and output of "Not equal! 10"

The output will be "Not equal 10". This illustrates that the Output +=10 calculation was never performed because processing stopped after the first operand was evaluated to be false. If you change the value of b1 to true processing occurs as you would expect and the output is "We are equal 20";.

Answer 46)

Objective 5.1)

```
2)j= i<<j;
4)j=i<<l;
```

Answer 47)

Objective 5.3)

4) 12

As well as the binary OR objective this questions requires you to understand the octal notaction which means that the leading letter zero (not the letter O)) means that the first 1 indicates the number contains one eight and nothing else. Thus this calculation in decimal mean

```
8|4
```

To convert this to binary means

```
1000
0100
----
1100
----
```

Which is 12 in decimal

The | bitwise operator means that for each position where there is a 1, results in a 1 in the same position in the answer.

Answer 48)

Objective 5.1)

2)s+=i;

Only a String acts as if the + operator were overloaded

Answer 49)

Objective 10.1)

Although the objectives do not specifically mention the need to understand the I/O Classes, feedback from people who have taken the exam indicate that you will get questions on this topic. As you will probably need to know this in the real world of Java programming, get familiar with the basics. I have assigned these questions to Objective 10.1 as that is a fairly vague objective.

1) File f = new File("/","autoexec.bat");
2) DataInputStream d = new DataInputStream(System.in);
3) OutputStreamWriter o = new OutputStreamWriter(System.out);

Option 4, with the RandomAccess file will not compile because the constructor must also be passed a mode parameter which must be either "r" or "rw"

Answer 50)

Objective 5.1)

```
1)o1=o2;
2)b=ob;
4)o1=b;
```

Answer 51)

Objective 5.4)

4) 10,0,20

In the call

another(v,i);

A reference to v is passed and thus any changes will be intact after this call.

Answer 52)

Objective 6.2)

1) public void amethod(String s, int i){}
4) public void Amethod(int i, String s) {}

Overloaded methods are differentiated only on the number, type and order of parameters, not on the return type of the method or the names of the parameters.

Answer 53)

Objective 6.2)

4)Base b = new Base(10);

Any call to this or super must be the first line in a constructor. As the method already has a call to this, no more can be inserted.

Answer 54)

Objective 4.1)

1)System.out.println(s);
4) System.out.println(iArgs);

A class within a method can only see final variables of the enclosing method. However it the normal visibility rules apply for variables outside the enclosing method.

Answer 55)

Objective 7.2)

1) yield()
2) sleep
4) stop()

Note, the methods stop and suspend have been deprecated with the Java2 release, and you may get questions on the exam that expect you to know this. Check out the Java2 Docs for an explanation

---

Answer 56)

Objective 10.1)

1) addElement

Answer 57)

Objective 4.1)

The import statement allows you to use a class directly instead of fully qualifying it with the full package name, adding more classess with the import statement does not cause a runtime performance overhad. An inner class can be defined with the private modifier.

3) An inner class can be defined with the protected modifier
4) An interface cannot be instantiated

Answer 58)

Objective 4.6)

1) mousePressed(MouseEvent e){}
4) componentAdded(ContainerEvent e){}

Answer 59)

Objective 10.1)

1) iterator
2) isEmpty
3) toArray

Answer 60)

Objective 7.3)

2) Ensures only one thread at a time may access a class or object

---

End of document