

# Java Programmer Certification Mock Exam

**Last Updated 4 June 1999**

This is a mock Exam based on the Objectives for the Sun Java Programmers Exam. It was created by Marcus Green ([mail@marcusgreen.co.uk](mailto: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.

I would appreciate it if you send me your score and the amount of java experience you have so I can gauge just how hard or easy it is. Send an email to [ExamScore@marcusgreen.co.uk](mailto:ExamScore@marcusgreen.co.uk).

Because the Objectives for the Java2 exam are very similar to the Objectives for the 1.1 exam, this test is appropriate for studying for the new exam. Questions from 45 onward may include objectives from Java2 exam. Updates will be posted to <http://www.marcusgreen.co.uk>. The answers contain references to approximately which objective the question relates to.

I am creating a new Mock exam based from the ground up on the Java2 Certification Objectives. It is under construction at the moment and only has a few question so far. Check it out at <http://www.software.u-net.com/javaexam/exam2.htm>

If you have your own ideas for questions, send them to me with answers and explanations. If they look good I'll post them and include a credit with your name and email address (if you like). Each question may have one or more correct answers.

## Questions

---

### Question 1)

Which of the following lines will compile without warning or error.

- 1) float f=1.3;
- 2) char c="a";
- 3) byte b=257;
- 4) boolean b=null;
- 5) int i=10;

[answer to Question 1](#)

---

### Question 2)

What will happen if you try to compile and run the following code

```
public class MyClass {  
    public static void main(String arguments[]) {  
        amethod(arguments);  
    }  
    public void amethod(String[] arguments) {  
        System.out.println(arguments);  
        System.out.println(arguments[1]);  
    }  
}
```

- 1) error Can't make static reference to void amethod.
- 2) error method main not correct
- 3) error array must include parameter
- 4) amethod must be declared with String

[answer to Question 2](#)

---

### Question 3)

Which of the following will compile without error

1)

```
import java.awt.*;  
package Mypackage;  
class Myclass {}
```

2)

```
package MyPackage;  
import java.awt.*;  
class MyClass{}
```

3)

```
/*This is a comment */  
package MyPackage;  
import java.awt.*;  
class MyClass{}
```

[answer to Question 3](#)

---

#### Question 4)

A byte can be of what size

- 1) -128 to 127
- 2)  $(-2^8)$  to  $(2^8)$
- 3) -255 to 256
- 4) depends on the particular implementation of the Java Virtual machine

[answer to Question 4](#)

---

#### Question 5)

What will be printed out if this code is run with the following command line?

```
java myprog good morning

public class myprog{

    public static void main(String argv[])

    {

        System.out.println(argv[2])

    }

}
```

- 1) myprog
- 2) good
- 3) morning
- 4) Exception raised: "java.lang.ArrayIndexOutOfBoundsException: 2"

[answer to Question 5](#)

---

#### Question 6)

Which of the following are java reserved words?

- 1) if
- 2) then
- 3) goto
- 4) while
- 5) case

[answer to Question 6](#)

---

### Question 7)

Which of the following are legal identifiers

- 1) 2variable
- 2) variable2
- 3) \_whatavariabale
- 4) \_3\_
- 5) \$anothervar
- 6) #myvar

[answer to Question 7](#)

---

### Question 8)

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

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

- 1) Error Variable i may not have been initialized
- 2) null
- 3) 1
- 4) 0

[answer to Question 8](#)

---

**Question 9)**

What will happen if you try to compile and run the following code?

```
public class Q {  
    public static void main(String argv[]){  
        int anar[]=new int[]{1,2,3};  
        System.out.println(anar[1]);  
    }  
}
```

- 1) 1
- 2) Error anar is referenced before it is initialized
- 3) 2
- 4) Error: size of array must be defined

[answer to Question 9](#)

---

**Question 10)**

What will happen if you try to compile and run the following code?

```
public class Q {  
    public static void main(String argv[]){  
        int anar[]=new int[5];  
        System.out.println(anar[0]);  
    }  
}
```

- 1) Error: anar is referenced before it is initialized
- 2) null
- 3) 0
- 4) 5

[answer to Question 10](#)

---

**Question 11)**

What will be the result of attempting to compile and run the following code?

```
abstract class MineBase {  
    abstract void amethod();  
    static int i;  
}  
  
public class Mine extends MineBase {  
    public static void main(String argv[]){  
        int[] ar=new int[5];  
        for(i=0;i < ar.length;i++)  
            System.out.println(ar[i]);  
    }  
}
```

- 1) a sequence of 5 0's will be printed
- 2) Error: ar is used before it is initialized
- 3) Error Mine must be declared abstract
- 4) IndexOutOfBounds Error

[answer to Question 11](#)

---

### Question 12)

What will be printed out if you attempt to compile and run the following code ?

```
int i=1;  
  
switch (i) {  
    case 0:  
        System.out.println("zero");  
    break;  
    case 1:  
        System.out.println("one");  
    case 2:  
        System.out.println("two");  
    default:
```

```
        System.out.println("default");  
    }
```

- 1) one
- 2) one, default
- 3) one, two, default
- 4) default

[answer to Question 12](#)

---

### Question 13)

What will be printed out if you attempt to compile and run the following code?

```
int i=9;  
switch (i) {  
    default:  
        System.out.println("default");  
        case 0:  
            System.out.println("zero");  
            break;  
    case 1:  
        System.out.println("one");  
    case 2:  
        System.out.println("two");  
}
```

- 1) default
- 2) default, zero
- 3) error default clause not defined
- 4) no output displayed

[answer to Question 13](#)

---

### Question 14)

Which of the following lines of code will compile without error

- 1)

```
int i=0;
```

```
if(i) {  
    System.out.println("Hello");  
}
```

2)

```
boolean b=true;  
boolean b2=true;  
if(b==b2) {  
    System.out.println("So true");  
}
```

3)

```
int i=1;  
int j=2;  
if(i==1 || j==2)  
    System.out.println("OK");
```

4)

```
int i=1;  
int j=2;  
if(i==1 & j==2)  
    System.out.println("OK");
```

[answer to Question 14](#)

---

### Question 15)

What will be output if you try to compile and run the following code, but there is no file called Hello.txt in the current directory?.

```
import java.io.*;  
public class Mine {  
    public static void main(String argv[]){  
        Mine m=new Mine();  
        System.out.println(m.amethod());  
    }  
}
```



```
public int amethod() {  
    try {  
        FileInputStream dis=new FileInputStream("Hello.txt");  
    }catch (FileNotFoundException fne) {  
        System.out.println("No such file found");  
        return -1;  
    }catch(IOException ioe) {  
    } finally{  
        System.out.println("Doing finally");  
    }  
    return 0;  
}
```

- 1) No such file found
- 2) No such file found ,-1
- 3) No such file found, Doing finally, -1
- 4) 0

[answer to Question 15](#)

---

### Question 16)

What tags are mandatory when creating HTML to display an applet

- 1) name, height, width
- 2) code, name
- 3) codebase, height, width
- 4) code, height, width

[answer to Question 16](#)

---

### Question 17)

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

- 1) Compile and run without error
- 2) Compile time Exception

### 3) Runtime Exception

```
class Base {}

class Sub extends Base {}

class Sub2 extends Base {}

public class CEx{

    public static void main(String argv[]){

        Base b=new Base();

        Sub s=(Sub) b;

    }

}
```

[answer to Question 17](#)

---

### Question 18)

If the following HTML code is used to display the applet in the code MgAp what will be displayed at the console?

- 1) Error: no such parameter
- 2) 0
- 3) null
- 4) 30

```
<applet name=MgAp code=MgAp.class height=400 width=400 parameter HowOld=30 >
</applet>
```

```
import java.applet.*;

import java.awt.*;

public class MgAp extends Applet{

    public void init(){

        System.out.println(getParameter("age"));

    }

}
```

[answer to Question 18](#)

---

**Question 19)**

You are browsing the Java HTML documentation for information on the `java.awt.TextField` component. You want to create Listener code to respond to focus events. The only Listener method listed is `addActionListener`. How do you go about finding out about Listener methods?

- 1) Define your own Listener interface according to the event to be tracked
- 2) Use the search facility in the HTML documentation for the listener needed
- 3) Move up the hierarchy in the HTML documentation to locate methods in base classes
- 4) Subclass `awt.event` with the appropriate Listener method

[answer to Question 19](#)

---

**Question 20)**

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

```
//Code start

import java.awt.*;

public class Butt extends Frame{

    public static void main(String argv[]){

        Butt MyBut=new Butt();

    }

    Butt(){

        Button HelloBut=new Button("Hello");

        Button ByeBut=new Button("Bye");

        add(HelloBut);

        add(ByeBut);

        setSize(300,300);

        setVisible(true);

    }

}

//Code end
```

- 1) Two buttons side by side occupying all of the frame, Hello on the left and Bye on

the right

- 2) One button occupying the entire frame saying Hello
- 3) One button occupying the entire frame saying Bye
- 4) Two buttons at the top of the frame one saying Hello the other saying Bye

[answer to Question 20](#)

---

### Question 21)

What will be output by the following code?

```
public class MyFor{  
    public static void main(String argv[]){  
        int i;  
        int j;  
        outer:  
        for (i=1;i <3;i++)  
            inner:  
            for(j=1; j<3; j++) {  
                if (j==2)  
                    continue outer;  
                System.out.println("Value for i=" + i + " Value for j=" +j);  
            }  
        }  
    }  
}
```

- 1) Value for i=1 value for j=1
- 2) value for i=2 value for j=1
- 3) value for i=2 value for j=2
- 4 value for i=3 value for j=1

[answer to Question 21](#)

---

### Question 22)

If g is a graphics instance what will the following code draw on the screen?.

```
g.fillArc(45, 90, 50, 50, 90, 180);
```

- 1) An arc bounded by a box of height 45, width 90 with a centre point of 50,50, starting at an angle of 90 degrees traversing through 180 degrees counter clockwise.
- 2) An arc bounded by a box of height 50, width 50, with a centre point of 45,90 starting at an angle of 90 degrees traversing through 180 degrees clockwise.
- 3) An arc bounded by a box of height 50, width 50, with a top left at coordinates of 45, 90, starting at 90 degrees and traversing through 180 degrees counter clockwise.
- 4) An arc starting at 45 degrees, traversing through 90 degrees clockwise bounded by a box of height 50, width 50 with a centre point of 90, 180.

[answer to Question 22](#)

---

### Question 23)

Which of the following methods can be legally inserted in place of the comment //Method Here ?

```
class Base{  
    public void amethod(int i) { }  
}  
  
public class Scope extends Base{  
    public static void main(String argv[]){  
        }  
    //Method Here  
}
```

- 1) void amethod(int i) throws Exception { }
- 2) void amethod(long i) throws Exception { }
- 3) void amethod(long i){ }
- 4) public void amethod(int i) throws Exception { }

[answer to Question 23](#)

---

### Question 24)

Which of the following will output -4.0

- 1) System.out.println(Math.floor(-4.7));
- 2) System.out.println(Math.round(-4.7));
- 3) System.out.println(Math.ceil(-4.7));
- 4) System.out.println(Math.Min(-4.7));

[answer to Question 24](#)

---

### Question 25)

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

```
Integer ten=new Integer(10);  
  
Long nine=new Long (9);  
  
System.out.println(ten + nine);  
  
int i=1;  
  
System.out.println(i + ten);
```

- 1) 19 followed by 20
- 2) 19 followed by 11
- 3) Error: Can't convert java lang Integer
- 4) 10 followed by 1

[answer to Question 25](#)

---

### Question 26)

If you run the code below, what gets printed out?

```
String s=new String("Bicycle");  
  
int iBegin=1;  
  
char iEnd=3;  
  
System.out.println(s.substring(iBegin,iEnd));
```

- 1) Bic
- 2) ic
- 3) icy
- 4) error: no method matching substring(int,char)

[answer to Question 26](#)

---

**Question 27)**

If you wanted to find out where the position of the letter v (ie return 2) in the string s containing "Java", which of the following could you use?

- 1) mid(2,s);
- 2) charAt(2);
- 3) s.indexOf('v');
- 4) indexOf(s,'v');

[answer to Question 27](#)

---

**Question 28)**

Given the following declarations

```
String s1=new String("Hello")  
String s2=new String("there");  
String s3=new String();
```

Which of the following are legal operations?

- 1) s3=s1 + s2;
- 2) s3=s1-s2;
- 3) s3=s1 & s2
- 4) s3=s1 && s2

[answer to Question 28](#)

---

**Question 29)**

What is the result of the following operation?

```
System.out.println(4 | 3);
```

- 1) 6
- 2) 0
- 3) 1
- 4) 7

[answer to Question 29](#)

---

**Question 30)**

```
public class MyClass1 {  
  
    public static void main(String argv[]){ }  
  
    /*Modifier at XX */ class MyInner {}  
  
}
```

What modifiers would be legal at XX in the above code?

- 1) public
- 2) private
- 3) static
- 4) friend

[answer to Question 30](#)

---

**Question 31)**

How would you go about opening an image file called MyPicture.jpg

- 1) Graphics.getGraphics("MyPicture.jpg");
- 2) Image image=Toolkit.getDefaultToolkit().getImage("MyPicture.jpg");
- 3) Graphics.openImage("MyPicture");
- 4) Image m=new Image("MyPicture");

[answer to Question 31](#)

---

**Question 32)**

An Applet has its Layout Manager set to the default of FlowLayout. What code would be correct to change to another Layout Manager.

- 1) setLayoutManager(new GridLayout());
- 2) setLayout(new GridLayout(2,2));
- 3) setGridLayout(2,2,))
- 4) setBorderLayout();

[answer to Question 32](#)



---

**Question 33)**

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

- 1) It will compile and the run method will print out the increasing value of i.
- 2) It will compile and calling start will print out the increasing value of i.
- 3) The code will cause an error at compile time.
- 4) Compilation will cause an error because while cannot take a parameter of true.

```
class Background implements Runnable{  
    int i=0;  
    public int run(){  
        while(true){  
            i++;  
            System.out.println("i="+i);  
        } //End while  
    } //End run  
} //End class
```

[answer to Question 33](#)

---

**Question 34)**

You have created an applet that draws lines. You have overridden the paint operation and used the graphics drawLine method, and increase one of its parameters to multiple lines across the screen. When you first test the applet you find that the news lines are redrawn, but the old lines are erased. How can you modify your code to allow the old lines to stay on the screen instead of being cleared.

- 1) Override repaint thus

```
public void repaint(Graphics g){  
    paint(g);  
}
```

- 2)Override update thus

```
public void update(Graphics g) {  
    paint(g);  
}
```

```
}
```

- 3) turn off clearing with the method setClear();
- 4) Remove the drawing from the paint Method and place in the calling code

[answer to Question 34](#)

---

### Question 35)

What will be the result when you attempt to compile and run the following code?.

```
public class Conv{  
    public static void main(String argv[]){  
        Conv c=new Conv();  
        String s=new String("ello");  
        c.amethod(s);  
    }  
    public void amethod(String s){  
        char c='H';  
        c+=s;  
        System.out.println(c);  
    }  
}
```

- 1) Compilation and output the string "Hello"
- 2) Compilation and output the string "ello"
- 3) Compilation and output the string elloH
- 4) Compile time error

[answer to Question 35](#)

---

### Question 36)

Given the following code, what test would you need to put in place of the comment line?

```
//place test here
```

to result in an output of

Equal

```
public class EqTest{  
    public static void main(String argv[]){  
        EqTest e=new EqTest();  
    }  
    EqTest(){  
        String s="Java";  
        String s2="java";  
        //place test here {  
            System.out.println("Equal");  
        }else  
        {  
            System.out.println("Not equal");  
        }  
    }  
}
```

- 1) if(s==s2)
- 2) if(s.equals(s2))
- 3) if(s.equalsIgnoreCase(s2))
- 4)if(s.noCaseMatch(s2))

[answer to Question 36](#)

---

### Question 37)

Given the following code

```
import java.awt.*;  
public class SetF extends Frame{  
    public static void main(String argv[]){  
        SetF s=new SetF();  
        s.setSize(300,200);  
    }  
}
```

```
s.setVisible(true);  
  
}  
  
}
```

How could you set the frame surface color to pink

- 1)s.setBackground(Color.pink);
- 2)s.setColor(PINK);
- 3)s.Background(pink);
- 4)s.color=Color.pink

[answer to Question 37](#)

---

### Question 38)

How can you change the current working directory using an instance of the File class called FileName?

- 1) FileName.chdir("DirName")
- 2) FileName.cd("DirName")
- 3) FileName.cwd("DirName")
- 4) The File class does not support directly changing the current directory.

[answer to Question 38](#)

---

### Question 39)

If you create a TextField with a constructor to set it to occupy 5 columns, what difference will it make if you use it with a proportional font (ie Times Roman) or a fixed pitch typewriter style font (Courier).

- 1)With a fixed font you will see 5 characters, with a proportional it will depend on the width of the characters
- 2)With a fixed font you will see 5 characters,with a proportional it will cause the field to expand to fit the text
- 3)The columns setting does not affect the number of characters displayed
- 4)Both will show exactly 5 characters

[answer to Question 39](#)

---

### Question 40)

Given the following code how could you invoke the Base constructor that will print out the string

"base constructor";

```
class Base{
    Base(int i){
        System.out.println("base constructor");
    }
    Base(){
    }
}

public class Sup extends Base{
    public static void main(String argv[]){
        Sup s= new Sup();
        //One
    }
    Sup()
    {
        //Two
    }
    public void derived()
    {
        //Three
    }
}
```

- 1) On the line After //One put Base(10);
- 2) On the line After //One put super(10);
- 3) On the line After //Two put super(10);
- 4) On the line After //Three put super(10);

[answer to Question 40](#)

---

### Question 41)

Given the following code what will be output?

```
public class Pass{
```

```
static int j=20;

public static void main(String argv[]){

    int i=10;

    Pass p = new Pass();

    p.amethod(i);

    System.out.println(i);

    System.out.println(j);

}

public void amethod(int x){

    x=x*2;

    j=j*2;

}

}
```

- 1) Error: amethod parameter does not match variable
- 2) 20 and 40
- 3) 10 and 40
- 4) 10, and 20

[answer to Question 41](#)

---

### Question 42)

What code placed after the comment //For loop would populate the elements of the array ia[] with values of the variable i.?

```
public class Lin{

public static void main(String argv[]){

    Lin l = new Lin();

    l.amethod();

}

public void amethod(){

    int ia[] = new int[4];

                                //Start For loop

                                {

        ia[i]=i;
```

```
        System.out.println(ia[i]);  
    }  
}  
  
}
```

- 1) for(int i=0; i < ia.length(); i++)
- 2) for (int i=0; i< ia.length(); i++)
- 3) for(int i=1; i < 4; i++)
- 4) for(int i=0; i< ia.length;i++)

[answer to Question 42](#)

---

### Question 43)

What will be the result when you try to compile and run the following code?

```
private class Base{  
    Base() {  
        int i = 100;  
        System.out.println(i);  
    }  
}  
  
public class Pri extends Base{  
    static int i = 200;  
    public static void main(String argv[]){  
        Pri p = new Pri();  
        System.out.println(i);  
    }  
}
```

- 1) Error at compile time
- 2) 200
- 3) 100 followed by 200
- 4) 100

[answer to Question 43](#)

---

**Question 44)**

What will the following code print out?

```
public class Oct{  
    public static void main(String argv[]){  
        Oct o = new Oct();  
        o.amethod();  
    }  
    public void amethod(){  
        int oi= 012;  
        System.out.println(oi);  
    }  
}
```

- 1)12
- 2)012
- 3)10
- 4)10.0

[answer to Question 44](#)

---

**Question 45**

What will happen when you try compiling and running this code?

```
public class Ref{  
    public static void main(String argv[]){  
        Ref r = new Ref();  
        r.amethod(r);  
    }  
    public void amethod(Ref r){  
        int i=99;  
        multi(r);  
        System.out.println(i);  
    }  
}
```



```
        public void multi(Ref r){  
            r.i = r.i*2;  
        }  
    }  
}
```

- 1) Error at compile time
- 2) An output of 99
- 3) An output of 198
- 4) An error at runtime

[answer to Question 45](#)

---

### Question 46)

You need to create a class that will store a unique object elements. You do not need to sort these elements but they must be unique.

What interface might be most suitable to meet this need?

- 1)Set
- 2)List
- 3)Map
- 4)Vector

[answer to Question 46](#)

---

### Question 47)

Which of the following will successfully create an instance of the Vector class and add an element?

- 1) Vector v=new Vector(99);  
v[1]=99;
- 2) Vector v=new Vector();  
v.addElement(99);
- 3) Vector v=new Vector();  
v.add(99);

```
4 Vector v=new Vector(100);  
v.addElement("99");
```

[answer to Question 47](#)

---

### Question 48)

You have created a simple Frame and overridden the paint method as follows

```
public void paint(Graphics g){  
    g.drawString("Dolly", 50, 10);  
}
```

What will be the result when you attempt to compile and run the program?

- 1) The string "Dolly" will be displayed at the centre of the frame
- 2) An error at compilation complaining at the signature of the paint method
- 3) The lower part of the word Dolly will be seen at the top of the form, with the top hidden.
- 4) The string "Dolly" will be shown at the bottom of the form

[answer to Question 48](#)

---

### Question 49)

What will be the result when you attempt to compile this program?

```
public class Rand{  
    public static void main(String argv[]){  
        int iRand;  
        iRand = Math.random();  
        System.out.println(iRand);  
    }  
}
```

- 1) Compile time error referring to a cast problem
- 2) A random number between 1 and 10
- 3) A random number between 0 and 1
- 4) A compile time error about random being an unrecognised method

[answer to Question 49](#)

---

**Question 50)**

Given the following code

```
import java.io.*;

public class Th{

    public static void main(String argv[]){

        Th t = new Th();

        t.amethod();

    }

    public void amethod(){

        try{

            ioCall();

        }catch(IOException ioe){}

    }

}
```

What code would be most likely for the body of the ioCall method

- 1) 

```
public void ioCall ()throws IOException{

    DataInputStream din = new DataInputStream(System.in);

    din.readChar();

}
```
- 2) 

```
public void ioCall ()throw IOException{

    DataInputStream din = new DataInputStream(System.in);

    din.readChar();

}
```
- 3) 

```
public void ioCall (){

    DataInputStream din = new DataInputStream(System.in);

    din.readChar();

}
```
- 4) 

```
public void ioCall throws IOException(){

    DataInputStream din = new DataInputStream(System.in);
```

```
din.readChar();  
}
```

[answer to Question 50](#)

---

### Question 51)

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

```
public class Scope{  
    private int i;  
  
    public static void main(String argv[]){  
        Scope s = new Scope();  
        s.amethod();  
    }//End of main  
  
    public static void amethod(){  
        System.out.println(i);  
    }//end of amethod  
}  
//End of class
```

- 1) A value of 0 will be printed out
- 2) Nothing will be printed out
- 3) A compile time error
- 4) A compile time error complaining of the scope of the variable i

[answer to Question 51](#)

---

### Question 52)

You want to lay out a set of buttons horizontally but with more space between the first button and the rest. You are going to use the GridBagLayout manager to control the way the buttons are set out. How will you modify the way the GridBagLayout acts in order to change the spacing around the first button?

- 1) Create an instance of the GridBagConstraints class, call the weightx() method and then pass the GridBagConstraints instance with the component to the setConstraints method of the GridBagLayout class.
- 2) Create an instance of the GridBagConstraints class, set the weightx field and then pass the GridBagConstraints instance with the component to the setConstraints method of the GridBagLayout

class.

- 3) Create an instance of the GridBagLayout class, set the weightx field and then call the setConstraints method of the GridBagLayoutClass with the component as a parameter.
- 4) Create an instance of the GridBagLayout class, call the setWeightx() method and then pass the GridBagConstraints instance with the component to the setConstraints method of the GridBagLayout class.

[answer to Question 52](#)

---

### Question 53)

Which of the following can you perform using the File class?

- 1) Change the current directory
- 2) Return the name of the parent directory
- 3) Delete a file
- 4) Find if a file contains text or binary information

[answer to Question 53](#)

---

### Question 54)

Which of the following code fragments will compile without error

1)

```
public void paint(Graphics g){  
    int polyX[] = {20,150,150};  
    int polyY[] = {20,20,120};  
    g.drawPolygon(polyX, polyY,3);  
}
```

2)

```
public void paint(Graphics g){  
    int polyX[] = {20,150,150};  
    int polyY[] = {20,20,120};  
    g.drawPolygon(polyX, polyY);  
}
```

3)

```
public void paint(Graphics g){  
    int polyX[3] = {20,150,150};  
    int polyY[3]= {20,20,120};  
    g.drawPolygon(polyX, polyY,3);  
}
```

4)

```
public void paint(Graphics g){  
    int polyX[] = {20,150,150};  
    int polyY[] = {20,20,120};  
    drawPolygon(polyX, polyY);  
}
```

[answer to Question 54](#)

---

### Question 55)

You are concerned about that your program may attempt to use more memory than is available. To avoid this situation you want to ensure that the Java Virtual Machine will run its garbage collection just before you start a complex routine. What can you do to be certain that garbage collection will run when you want .

- 1) You cannot be certain when garbage collection will run
- 2) Use the Runtime.gc() method to force garbage collection
- 3) Ensure that all the variables you require to be garbage collected are set to null
- 4) Use the System.gc() method to force garbage collection

[answer to Question 55](#)

---

### Question 56)

You are using the GridBagLayout manager to place a series of buttons on a Frame. You want to make the size of one of the buttons bigger than the text it contains. Which of the following will allow you to do that?

- 1) The GridBagLayout manager does not allow you to do this
- 2) The setFill method of the GridBagLayout class
- 3) The setFill method of the GridBagConstraints class
- 4) The fill field of the GridBagConstraints class

[answer to Question 56](#)

---

### Question 57)

Which of the following most closely describes a bitset collection?

- 1) A class that contains groups of unique sequences of bits
- 2) A method for flipping individual bits in instance of a primitive type
- 3) An array of boolean primitives that indicate zeros or ones
- 4) A collection for storing bits as on-off information, like a vector of bits

[answer to Question 57](#)

---

### Question 58)

You have these files in the same directory. What will happen when you attempt to compile and run Class1.java if you have not already compiled Base.java

```
//Base.java

package Base;

class Base{

    protected void amethod(){

        System.out.println("amethod");

    }//End of amethod

}//End of class base

package Class1;

//Class1.java

public class Class1 extends Base{

    public static void main(String argv[]){

        Base b = new Base();

        b.amethod();

    }//End of main

}//End of Class1
```

- 1) Compile Error: Methods in Base not found
- 2) Compile Error: Unable to access protected method in base class
- 3) Compilation followed by the output "amethod"
- 4) Compile error: Superclass Class1.Base of class Class1.Class1 not found

[answer to Question 58](#)

---

### Question 59)

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

```
class Base{  
    private void amethod(int iBase){  
        System.out.println("Base.amethod");  
    }  
}  
  
class Over extends Base{  
  
    public static void main(String argv[]){  
        Over o = new Over();  
        int iBase=0;  
        o.amethod(iBase);  
    }  
    public void amethod(int iOver){  
        System.out.println("Over.amethod");  
    }  
}
```

- 1) Compile time error complaining that Base.amethod is private
- 2) Runtime error complaining that Base.amethod is private
- 3) Output of Base.amethod
- 4) Output of Over.amethod()



[answer to Question 59](#)

---

### Question 60)

You are creating an applet with a Frame that contains buttons. You are using the GridBagLayout manager and you have added Four buttons. At the moment the buttons appear in the centre of the frame from left to right. You want them to appear one on top of the other going down the screen. What is the most appropriate way to do this.

- 1) Set the gridy value of the GridBagConstraints class to a value increasing from 1 to 4
- 2) set the fill value of the GridBagConstraints class to VERTICAL
- 3) Set the ipady value of the GridBagConstraints class to a value increasing from 0 to 4
- 4) Set the fill value of the GridBagLayout class to GridBag.VERTICAL

[answer to Question 60](#)

---

## Answers

---

### Answer 1)

[Back to question 1\)](#)

Objective 2.6

5) `int i=10;`

explanation:

1) `float f=1.3;`

Will not compile because the default type of a number with a floating point component is a double. This would compile with a cast as in

`float f=(float) 1.3`

2) `char c="a";`

Will not compile because a char (16 bit unsigned integer) must be defined with single quotes. This would compile if it were in the form

`char c=' a' ;`

3) `byte b=257;`

Will not compile because a byte is eight bits. Take of one bit for the sign component you can define numbers between

-127 to +127

4) a boolean value can either be true or false, null is not allowed.

---

### Answer 2)

[Back to question 2\)](#)

#### Objective 3.1

1) Can't make static reference to void method.

Because main is defined as static you need to create an instance of the class in order to call any non-static methods. Thus a typical way to do this would be.

```
MyClass m=new MyClass();
```

```
m.amethod();
```

Answer 2 is an attempt to confuse because the convention is for a main method to be in the form

```
String argv[]
```

That argv is just a convention and any acceptable identifier for a string array can be used. Answers 3 and 4 are just nonsense.

---

### Answer 3)

[back to Question 3\)](#)

#### Objective 1.2

2 and 3 will compile without error.

1 will not compile because any package declaration must come before any other code. Comments may appear anywhere.

---

### Answer 4)

[Back to question 4\)](#)

Objective 1.9)

1) A byte is a signed 8 bit integer.

---

**Answer 5)**

[Back to question 5\)](#)

Objective 1.4)

4) Exception raised: "java.lang.ArrayIndexOutOfBoundsException: 2"

Unlike C/C++ java does not start the parameter count with the program name. It does however start from zero. So in this case zero starts with good, morning would be 1 and there is no parameter 2 so an exception is raised.

---

**Answer 6)**

[Back to question 6\)](#)

Objective 1.5)

- 1) if
- 3) goto
- 4) while
- 5) case

*then* is not a Java keyword, though if you are from a VB background you might think it was. Goto is a reserved word in Java.

---

**Answer 7)**

[Back to Question 7\)](#)

Objective 1.10)

- 2) variable2
- 3) \_whatavariabale
- 4) \_3\_
- 5) \$anothervar

An identifier can begin with a letter (most common) or a dollar sign(\$) or an underscore(\_). An identifier cannot start with anything else such as a number, a hash, # or a dash -. An identifier cannot have a dash in its body, but it may have an underscore \_. Choice 4) \_3\_ looks strange but it is an

acceptable, if unwise form for an identifier.

---

**Answer 8)**

[Back to Question 8\)](#)

Objective 1.6)

4) 0

Class level variables are always initialised to default values. In the case of an int this will be 0. Method level variables are not given default values and if you attempt to use one before it has been initialised it will cause the

Error Variable i may not have been initialized

type of error.

---

**Answer 9)**

[Back to Question 9\)](#)

Objective 1.7,3.4)

3 ) 2

No error will be triggered.

Like in C/C++, arrays are always referenced from 0. Java allows an array to be populated at creation time. The size of array is taken from the number of initializers. If you put a size within any of the square brackets you will get an error.

---

**Answer 10)**

[Back to question 10\)](#)

Objective 1.7)

3) 0

Arrays are always initialised when they are created. As this is an array of ints it will be initialised with zeros.

---

**Answer 11)**

[Back to Question 11\)](#)

Objective 3.6

3) Error Mine must be declared abstract

A class that contains an abstract method must itself be declared as abstract. It may however contain non abstract methods. Any class derived from an abstract class must either define all of the abstract methods or be declared abstract itself.

---

**Answer 12)**

[Back to Question 12\)](#)

Objective 4.1)

3) one, two, default

Code will continue to fall through a case statement until it encounters a break.

---

**Answer 13)**

[Back to Question 13\)](#)

Objective 4.1)

2) default, zero

Although it is normally placed last the default default statement does not have to be the last item as you fall through the case block. Because there is no case label found matching the expression the default label is executed and the code continues to fall through until it encounters a break.

---

**Answer 14)**

[Back to Question 14\)](#)

Objective 4.2,

2,3

Example 1 will not compile because if must always test a boolean. This can catch out C/C++ programmers who expect the test to be for either 0 or not 0.

---

**Answer 15)**

[Back to Question 15\)](#)

Objective 4.5)

3) No such file found, doing finally, -1

The no such file found message is to be expected, however you can get caught out if you are not aware that the finally clause is almost always executed, even if there is a return statement.

---

**Answer 16)**[Back to Question 16\)](#)

Objective Unknown

4) code, height, width

---

**Answer 17)**[Back to Question 17\)](#)

Objective 5.8 (sort of)

3) Runtime Exception

Without the cast to sub you would get a compile time error. The cast tells the compiler that you really mean to do this and the actual type of b does not get resolved until runtime. Casting down the object hierarchy as the compiler cannot be sure what has been implemented in descendent classes. Casting up is not a problem because sub classes will have the features of the base classes. This can feel counter intuitive if you are aware that with primitives casting is allowed for widening operations (ie byte to int).

---

**Answer 18)**[Back to question 18\)](#)

Objective unknown

3) null

If a parameter is not available the applet will still run, but any attempt to access the parameter will return a null.

---

**Answer 19)**[Back to Question 19\)](#)

Objective 1.1)

3) Move up the hierarchy in the HTML documentation to locate methods in base classes

The documentation created by JavaDoc is based on tags placed into the sourcecode. The convention for documentation is that methods and fields of ancestors are not duplicated in sub classes. So if you are looking for something and it does not appear to be there, you move up the class hierarchy to find it.

---

**Answer 20)**[Back to Question 20\)](#)

Objective 10.4)

3) One button occupying the entire frame saying Bye

The default layout manager for a Frame is a border layout. If directions are not given (ie North, South, East or West), any button will simply go in the centre and occupy all the space. An additional button will simply be placed over the previous button. What you would probably want in a real example is to set up a flow layout as in

`setLayout(new FlowLayout());` which would.

Applets and panels have a default FlowLayout manager

---

**Answer 21)**[Back to Question 21\)](#)

Objective 4.4)

1,2

Value for i=1 Value for j=1

Value for i=2 Value for j=1

The statement `continue outer` causes the code to jump to the label `outer` and the for loop increments to the next number.

---

**Answer 22)**[Back to Question 22\)](#)

Objective 9.5)

3) An arc bounded by a box of height 50, width 50, with a top left at coordinates of 45, 90, starting at 90 degrees and traversing through 180 degrees counter clockwise.

`fillArc(int x, int y, int width, int height, int startDegrees, int arcDegrees)`

The `fillArc` function draws an arc in a box with a top left at coordinates X & Y.  
If the `ArcDegrees` is a positive number the arc is drawn counter clockwise.

---

**Answer 23)**

[Back to Question 23\)](#)

Objective 4.7)

2,4

Options 1, & 4 will not compile as they attempt to throw Exceptions not declared in the base class.  
Because options 2 and 4 take a parameter of type `long` they represent overloading not overriding and there is no such limitations on overloaded methods.

---

**Answer 24)**

[Back to Question 24\)](#)

Objective 8.1)

3) `System.out.println(Math.ceil(-4.7));`

Options 1 and 2 will produce -5 and option 4 will not compile because the `Min` method requires 2 parameters.

---

**Answer 25)**

[Back to Question 25\)](#)

Objective 2.2

3) Error: Cant convert java lang Integer

The wrapper classes cannot be used like primitives.

Wrapper classes have similar names to primitives but all start with upper case letters.  
Thus in this case we have `int` as a primitive and `Integer` as a wrapper. The objectives do not specifically mention the wrapper classes but don' t be surprised if they come up.

---

**Answer 26)**

[Back to Question 26\)](#)

Objective 8.2)



2) ic

This is a bit of a catch question. Anyone with a C/C++ background would figure out that addressing in strings starts with 0 so that 1 corresponds to i in the string Bicycle. The catch is that the second parameter returns the endcharacter minus 1. In this case it means instead of the "icy" being returned as intuition would expect it is only "ic".

---

**Answer 27)**

[Back to Question 27\)](#)

Objective 8.2)

3) s.indexOf(' v' );

charAt returns the letter at the position rather than searching for a letter and returning the position, MID is just to confuse the Basic Programmers, indexOf(s, ' v' ); is how some future VB/J++ nightmare hybrid, might perform such a calculation.

---

**Answer 28)**

Objective 2.2

[Back to Question 28](#)

1) s3=s1 + s2;

Java does not allow operator overloading as in C++, but for the sake of convenience the + operator is overridden for strings.

---

**Answer 29)**

[Back to Question 29\)](#)

Objective 2.5)

4) 7

The | is known as the Or operator, you could think of it as the either/or operator. Turning the numbers into binary gives

4=100

3=011

For each position, if either number contains a 1 the result will contain a result in that position. As

every position contains a 1 the result will be

111

Which is decimal 7.

---

**Answer 30)**

[Back to Question 30](#)

Objective 3.7)

1,2,3

public, private, static are all legal access modifiers for this inner class.

---

**Answer 31)**

[Back to Question 31](#)

Objective 9.6)

Opening an image file requires an Image object, The Image class has no constructor that takes the name of an image file . For an application (rather than an applet) an image is created using the Toolkit class as in option 2.

2) Image image=Toolkit.getDefaultToolkit().getImage("MyPicture.jpg");

---

**Answer 32)**

[Back to Question 32\)](#)

Objective 1.3)

2) setLayout(new GridLayout(2,2));

Changing the layout manager is the same for an Applet or an application. Answer 1 is wrong and implausible as a standard method is unlikely to have a name as long as setLayoutManager. Answers 3 and 4 are incorrect because changing the layout manager always requires an instance of one of the Layout Managers and these are bogus methods.

Instead of creating the anonymous instance of the Layout manager as in option 2 you can also create a named instance and pass that as a parameter. This is often what automatic code generators such as Borland/Inprise JBuilder do.

---

**Answer 33)**[Back to Question 33\)](#)

Objective 7.2)

3) The code will cause an error at compile time

The error is caused because run should have a void not an int return type.

Any class that implements an interface must create a method to match all of the methods in the interface. The Runnable interface has one method called run that has a void return type. The sun compiler gives the error

Method redefined with different return type: int run() was defined as void run();

---

**Answer 34)**[Back to Question 34\)](#)

Objective 11.1)

```
2) public void update(Graphics g) {  
    paint(g);  
}
```

If not overridden the update method clears the background and calls paint(); By overriding the update method, any previously drawn graphics will not be cleared. This is only a trivial way of preserving any graphics drawn. If the application is resized or the drawing area covered in some way the graphics will be cleared.

---

**Answer 35)**[Back to Question 35](#)

Objective 2.2

4) Compile time error

The only operator overloading offered by java is the + sign for the String class. A char is a 16 bit integer and cannot be concatenated to a string with the + operator.

---

**Answer 36)**

[Back to Question 36](#)

Objective 8.2)

3) if(s.equalsIgnoreCase(s2))

String comparison is case sensitive so using the equals string method will not return a match. Using the == operator just compares where memory address of the references and noCaseMatch was just something I made up to give me a fourth slightly plausible option.

---

**Answer 37)**

[Back to Question 37](#)

Objective 9.1)

1) s.setBackground(Color.pink);

For speakers of the more British spelt English note that there is no letter u in Color. Also the constants for colors are in lower case.

---

**Answer 38)**

[Back to Question 38\)](#)

Objective 13.1)

4) The File class does not support directly changing the current directory.

This seems rather surprising to me, as changing the current directory is a very common requirement. You may be able to get around this limitation by creating a new instance of the File class passing the new directory to the constructor as the path name.

---

**Answer 39)**

[Back to Question 39\)](#)

Objective 9.2)

1) With a fixed font you will see 5 characters, with a proportional it will depend on the width of the characters

With a proportional font the letter w will occupy more space than the letter i. So if you have all wide characters you may have to scroll to the right to see the entire text of a TextField.

---

**Answer 40)**

[Back to Question 40\)](#)

Objective 5.8

3) On the line After //Two put super(10);

Constructors can only be invoked from within constructors.

---

**Answer 41)**

[Back to Question 41\)](#)

Objective 2.8)

3) 10 and 40

when a parameter is passed to a method the method receives a copy of the value. The method can modify its value without affecting the original copy. Thus in this example when the value is printed out the method has not changed the value.

---

**Answer 42)**

[Back to Question 42](#)

Objective 3.3

4) for(int i=0; i< ia.length;i++)

Although you could control the looping with a literal number as with the number 4 used in sample 3, it is better practice to use the length property of an array. This provides against bugs that might result if the size of the array changes. This question also checks that you know that arrays starts from zero and not One.

---

**Answer 43)**

[Back to Question 43\)](#)

Objective 3.6 (maybe)

1) Error at compile time

This is a slightly sneaky one as it looks like a question about constructors, but it is attempting to test knowledge of the use of the private modifier. A top level class cannot be defined as private. If you didn't notice the modifier private, remember in the exam to be real careful to read every part of the question.

---

**Answer 44)**

[Back to Question 44](#)

Objective 1.11)

3)10

The name of the class might give you a clue with this question, Oct for Octal. Prefixing a number with a zero indicates that it is in Octal format. Thus when printed out it gets converted to base ten. 012 in octal means the first column from the right has a value of 2 and the next along has a value of one times eight. In decimal that adds up to 10.

---

**Answer 45)**

[Back to Question 45](#)

Objective 3.5)

1) Error at compile time

The variable i is created at the level of a method and will not be available inside the method multi.

---

**Answer 46)**

[Back to Question 46](#)

Java2 Objective 10.1)

1) Set

The Set interface ensures that its elements are unique, but does not order the elements. In reality you probably wouldn't create your own class using the Set interface. You would be more likely to use one of the JDK classes that use the Set interface such as `ArraySet`.

---

**Answer 47)**

[Back to Question 47](#)

Java2 Objective 10.1)

4) `Vector v=new Vector(100);`  
`v.addElement("99")`

A vector can only store objects not primitives. The parameter "99" for the `addElement` method passes a string object to the Vector. Option 1) creates a vector OK but then uses array syntax to attempt to assign a primitive. Option 2 also creates a vector then uses correct Vector syntax but falls over when

the parameter is a primitive instead of an object. Option 3 compounds the errors by using the fictitious add method.

---

### Answer 48)

Objective 9.5)

[Back to Question 48](#)

3) The lower part of the word Dolly will be seen at the top of the form

The Second parameter to the drawstring method indicates where the baseline of the string will be placed. Thus the 3rd parameter of 10 indicates the Y coordinate to be 10 pixels from the top of the Frame. This will result in just the bottom of the string Dolly showing up or possibly only the descending part of the letter y.

---

### Answer 49)

[Back to Question 49\)](#)

Objective 8.1)

1) Compile time error referring to a cast problem

This is a bit of a sneaky one as the Math.random method returns a pseudo random number between 0 and 1, and thus option 3 is a plausible answer. However the number returned is a double and so the compiler will complain that a cast is needed to convert a double to an int.

---

### Answer 50)

Objective 4.6)

[Back to question 50](#)

```
1) public void ioCall ()throws IOException{  
    DataInputStream din = new DataInputStream(System.in);  
    din.readChar();  
}
```

If a method might throw an exception it must either be caught within the method with a try/catch block, or the method must indicate the exception to any calling method by use of the throws statement in its declaration. Without this, an error will occur at compile time.

---

**Answer 51)**

Objective 3.10)

[Back to Question 51\)](#)

3) A compile time error

Because only one instance of a static method exists not matter how many instance of the class exists it cannot access any non static variables. The JVM cannot know which instance of the variable to access. Thus you will get an error saying something like

```
Can't make a static reference to a non static variable
```

---

**Answer 52)**

Java2 Objective 8.2)

[Back to Question 52\)](#)

2) Create an instance of the GridBagConstraints class, set the weightx field and then pass the GridBagConstraints instance with the component to the setConstraints method of the GridBagLayout class.

The Key to using the GridBagLayout manager is the GridBagConstraints class. This class is not consistent with the general naming conventions in the java API as you would expect that weightx would be *set* with a method, whereas it is a simple field (variable).

---

**Answer 53)**

Objective 13.1)

[Back to Question 53\)](#)

- 2) Return the name of the parent directory
- 3) Delete a file

It is surprising that you can' t change the current directory. If you need to do this, the best way seems to be to create a new instance of the File class and pass the new directory to the constructor. It is not so surprising that you can' t tell if a file contains text or binary information.

---

**Answer 54)**



Objective 9.5)

[Back to Question 54\)](#)

1)

```
public void paint(Graphics g){  
    int polyX[] = {20,150,150};  
    int polyY[] = {20,20,120};  
    g.drawPolygon(polyX, polyY,3);  
}
```

Drawpolygon takes three parameters, the first two are arrays of the X,Y coordinates and the final is n integer specifying the number of vertices (whatever they are).

---

**Answer 55)**

Objective 6.1)

[Back to Question 55\)](#)

1) You cannot be certain when garbage collection will run

Although there is a Runtime.gc(), this only suggests that the Java Virtual Machine does its garbage collection. You can never be certain when the garbage collector will run. Roberts and Heller is more specific about this than Boone. This uncertainty can cause consternation for C++ programmers who wish to run finalize methods with the same intent as they use destructor methods.

---

**Answer 56)**

Java2 Objective 8.2)

[Back to Question 56\)](#)

4) The fill field of the GridBagConstraints class

Unlike the GridLayout manager you can set the individual size of a control such as a button using the GridBagLayout manager. A little background knowledge would indicate that it should be controlled by a setSomethingOrOther method, but it isn't.

---

**Answer 57)**

Java2 Objective 10.1)

[Back to Question 57\)](#)

4) A collection for storing bits as on-off information, like a vector of bits

This is the description given to a bitset in Bruce Eckels "Thinking in Java" book. The reference to unique sequence of bits was an attempt to mislead because of the use of the word Set in the name bitset. Normally something called a set implies uniqueness of the members, but not in this context.

---

**Answer 58)**

[Back to Question 58\)](#)

Objective 3.10)

4)Compile error: Superclass Class1.Base of class Class1.Class1 not found

Using the package statement has an effect similar to placing a source file into a different directory. Because the files are in different packages they cannot see each other. The stuff about File1 not having been compiled was just to mislead, java has the equivalent of an "automake", whereby if it was not for the package statements the other file would have been automatically compiled.

---

**Answer 59)**

[Back to Question 59\)](#)

Objective 5.3)

4) Output of Over.amethod()

The names of parameters to an overridden method is not important.

---

**Answer 60)**

Java2 Objective 8.2)

[Back to Question 60\)](#)

1) Set the gridy value of the GridBagConstraint class to a value increasing from 1 to 4

Answer 4 is fairly obviously bogus as it is the GridBagConstraint class that does most of the magic in laying out components under the GridBagLayout manager. The fill value of the GridBagConstraint class controls the behavior inside its virtual cell and the ipady field controls the internal padding

around a component.

---

**This page has been accessed  times**