Cram Bible Lab

EasyJAVA Cram Bible

Sun Certified Java Programmer
Exam 310-022/310-025

(Release 1.0)

**** Single User License ****
This copy can be only used by yourself for educational purposes

Before you get started:

EasyJAVA Cram Bible includes questions for both 2 versions of the current Sun Java Certified Programmer tests:

1) **(310-022) SUN CERTIFIED PROGRAMMER FOR THE JAVA PLATFORM JDK 1.1**
2) **(310-025) SUN CERTIFIED PROGRAMMER FOR THE JAVA 2 PLATFORM**

Part 1 is aiming at the JAVA JDK1.1 test, and Part 2 is aiming at the JAVA 2 test. Part 3 includes questions both for these 2 tests. BTW, you could choose any version of the test to get SCJP certified.

For more details about the SCJP tests, you could go to:
http://suned.sun.com/certification/progdetails.html
For more details about how to take the SCJP tests in Chinese mainland, you could go to:
http://www.sun.com.cn/education/sun-auth.html

If you take the (310-025) JAVA2 test, we seriously recommend you study all the 3 parts thoroughly. Actually, most of the contents of the 2 version tests are just the basics of JAVA language, and they're identical.

Study hard, and you'll absolutely pass the SCJP test at the very first time efficiently.

(Part 1) SUN CERTIFIED PROGRAMMER FOR THE JAVA PLATFORM, JDK 1.1
(122 Questions and Explanations)

Question: 1
What is the permanent effect on the file system
of writing data to a new FileWriter("report"), given
the file report already exists?

A. The data is appended to the file.
B. The file is replaced with a new file.
C. A checked exception is raised as the file already exists.
D. A runtime exception is raised as the file already exists.
E. The data is written to random locations within the file.

Answer: B

Question: 2
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
A. Two buttons side by side occupying all of the frame, Hello on the left and Bye on the right
B. One button occupying the entire frame saying Hello
C. One button occupying the entire frame saying Bye
D. Two buttons at the top of the frame one saying Hello the other saying Bye

Answer:
C. One button occupying the entire frame saying Bye
Explanation:
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

Question: 3
Which Listener interfaces can you add to a TextArea object?

A. TextListener
B. ActionListener
C. MouseMotionListener
D. MouseListener
E. ComponentListener

Answer:
A., C., D., and E..
Explanation:
B. is not correct, because you cannot add an
ActionListener to a TextArea. (You can add an ActionListener to a TextField, however.)

Question: 4
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
         char c='H';
         c+=s;
         System.out.println(c);
      }
}
```

A. Compilation and output the string "Hello"
B. Compilation and output the string "ello"
C. Compilation and output the string elloH
D. Compile time error

Answer:
D. Compile time error

Explanation:
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.

Question: 5
An Applet has its Layout Manager set to the default of
FlowLayout. What code would be correct to change to
another Layout Manager?

A. setLayoutManager(new GridLayout());
B. setLayout(new GridLayout(2,2));
C. setGridLayout(2,2,))
D. setBorderLayout();
Question Help.
Answer:
B. setLayout(new GridLayout(2,2));

Explanation:

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 C. and D. 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 B. 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.

Question: 6
Which of the following are java reserved words?
Question Help.
Answers:
if
goto
while
case

Explanation:
Option:
then is not a Java keyword, though if you are from a VB background you might think it was.

Option:
goto IS a reserved word in Java.

Question: 7
Which interface implementations can you add as listeners for a TextField object?

A. ActionListener

B. FocusListener

C. MouseMotionListener

D. WindowListener

E. ContainerListener
Question Help.
Answer:
A., B., and C..

Explanation:
The answers d and e are not valid for TextField objects.

Question: 8
Given that the variable g references a valid Graphics object, what does the following statement do?

g.fillRect(2, 3, 10, 20);

A. draw the outline of a rectangle in the current background color

B. draw the outline of a rectangle in the current foreground color

C. fill in a rectangle using the current background color

D. fill in a rectangle using the current foreground color

E. fill in a rectangle in black

Question Help.

Answer:
D. is the only correct answer.

Explanation:
There is a method called drawRect() that draws the outline of a rectangle.

Question: 9
What will the user interface look like in an applet given the following init() method?

```
public void init() {
    setLayout(new BorderLayout());
    add("East", new Button("hello"));
}
```

A. Nothing will appear in the applet

B. A button will appear in the applet set in the exact center

C. A button will appear on the left side of the applet

D. A button will appear on the right side of the applet

E. A button will fill the entire applet

Question Help.

Answer:
D.

Explanation:
Try it and see!

Question: 10
A class design requires that a particular member variable must be accessible for direct access by any subclasses of this class, but otherwise not by classes which are not members of the same package. What should be done to achieve this?

A. The variable should be marked public

B. The variable should be marked private

C. The variable should be marked protected

D. The variable should have no special access modifier

E. The variable should be marked private and an accessor method provided
Question Help.
Answer:
C.

Question: 11
What appears in the standard output when the method named testing is invoked?

```
void testing() {
one:
two:
    for (int i = 0; i < 3; i++) {
three:
        for (int j = 10; j < 30; j+=10) {
            System.out.println(i + j);
            if (i > 2)
                continue one;
        }
    }
}
```

A. 10 and 20

B. 11 and 21

C. 12 and 22

D. 13 and 23

E. 30, 31, 32, 33
Question Help.
Answer:
A., B., and C.

Question: 12
Which methods may cause a thread to stop executing?

A. sleep();
B. stop();
C. yield();
D. wait();
E. notify();
F. notifyAll()
G. synchronized()
Question Help.
Answer:
A., B., C. and D.

Question: 13
Which of the following are legal identifiers?

A. 2variable
B. variable2
C. _whatavariable
D. _3_
E. $anothervar
F. #myvar
Question Help.
Answer:
variable2
_whatavariable
_3_
$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.

Question: 14
What can you write at the comment:
//A in the code below so that this program writes the word "running" to the standard output?

```
class RunTest implements Runnable {
    public static void main(String[] args) {
        RunTest rt = new RunTest();
        Thread t =new Thread(rt);
        //A
    }
    public void run() {
        System.out.println("running");
    }
    void go() {
        start(1);
    }
    void start(int i) {
    }
}
```

A. System.out.println("running");

B. rt.start();

C. rt.go();

D. rt.start(1);
Question Help.
Answer:
A. is the only correct answer.

Question: 15
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);
```

A. 19 followed by 20
B. 19 followed by 11
C. Error: Can't convert java lang Integer
D. 10 followed by 1
Question Help.
Answer:
C. Error: Cant convert java lang Integer

Explanation:
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.

Question: 16
What will the user interface look like in an applet
given the following init() method?

```
public void init() {
    setLayout(new BorderLayout());
    add(new Button("hello"));
}
```

A. Nothing will appear in the applet

B. A button will appear in the applet set in the exact center

C. A button will appear in the applet along the top and
centered horizontally

D. A button will appear in the top left corner

E. A button will fill the entire applet
Question Help.
Answer:
E.

Explanation:
Again, try it and see!

Question: 17
What is written to the standard output given the following statement:

System.out.println(4 & 7);

A. 4

B. 5

C. 6

D. 7

E. 0
Question Help.
Answer:
A.

Explanation:
The value of

4 & 7

can be seen clearly as binary digits:

100 &
111

100

or 4.

Question: 18
Which of the following correctly illustrate how an InputStreamReader can be created:

A. new InputStreamReader(new FileInputStream("data"));
B. new InputStreamReader(new FileReader("data"));
C. new InputStreamReader(new BufferedReader("data"));
D. new InputStreamReader("data");
E. new InputStreamReader(System.in);
Question Help.
Answer:
A. and E.

Question: 19
Which best describes the user interface of an applet given the following init() method:

```
public void init() {
    setLayout(new BorderLayout());
    add("North", new TextField(10));
    add("Center", new Button("help"));
}
```

A. The TextField object will be placed at the top of
the applet and will be 10 columns wide

B. The Button object will be centered in the applet
and will be just large enough to contain the text "help"

C. The Button object will be centered in the applet and
will start at the left edge of the applet, fit just under
the TextField object above it, and extend to the right
and bottom edge of the applet

D. The TextField object will be placed along the top of the
applet and will stretch from the left edge to the right edge

E. The placement of the Button object and the TextField object
depends on the overall size of the applet.
Question Help.
Answer:
C. and D.

Question: 20
Analyze this line of code:

```
if (5 & 7 > 0 && 5 | 2) System.out.println("true");
```

A. this line of code will not compile

B. this code will compile but nothing will appear in the
standard output

C. this code will compile and write the word "true" in the
standard output
Question Help.
Answer:
A.

Explanation:
The problem is in
5 | 2
An if() expression must resolve to true or false.

Question: 21
What must be true for the RunHandler class so that
instances of RunHandler can be used as written in
the code below:

```
class Test {
    public static void main(String[] args) {
        Thread t = new Thread(new RunHandler());
        t.start();
    }
}
```

A. RunHandler must implement the java.lang.Runnable
interface.

B. RunHandler must extend the Thread class.

C. RunHandler must provide a run() method declared
as public and returning void.

D. RunHandler must provide an init() method.
Question Help.
Answer:
A. and C. are the correct answers.

Explanation:
The answer B. is an option, but it does not have to
be so -- RunHandler could extend Thread, which would
then indicate that it implements the Runnable interface -- but
it does not need to extend Thread to implement Runnable.

Question: 22
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;
     }
}
```

A. Error: amethod parameter does not match variable
B. 20 and 40
C. 10 and 40
D. 10, and 20
Question Help.
Answer:
C. 10 and 40

Explanation:
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.

Question: 23
Which are keywords in Java?

A. sizeof

B. abstract

C. native

D. NULL

E. BOOLEAN
Question Help.
Answer:
B. and C.

Question: 24
To determine if you can invoke addContainerListener() for a component referenced using a variable named c, which expression(s) can you evaluate that will give you a true or false answer to this questions?

A. c == Container

B. c.equals(Class.Container)

C. c instanceof Container

D. c instanceof Component

E. c implements Container
Question Help.
Answer:
C. is the only correct answer.

Question: 25
What is the result of executing the following Java class:

```
import java.awt.*;

public class FrameTest extends Frame {

    public FrameTest() {
        add (new Button("First"));
        add (new Button("Second"));
        add (new Button("Third"));
        pack();
        setVisible(true);
    }

    public static void main(String args []) {
        new FrameTest();
    }
}
```

Select from the following options:

A. Nothing happens.
B. Three buttons are displayed across a window.

C. A runtime exception is generated (no layout manager specified).
D. Only the "first" button is displayed.
E. Only the "second" button is displayed.
F. Only the "third" button is displayed.
Question Help.
Answer:
F.

Question: 26
Which of the following code fragments will compile without error?

A.

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

B.

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

C.

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

D.

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

Question Help.
Answer:
A.

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

Explanation:
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).

Question: 27
You have created an applet that draws lines. You have overriden 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.

A. Override repaint thus

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

B. Override update thus

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

C. turn off clearing with the method setClear();

D. Remove the drawing from the paint Method
and place in the calling code
Question Help.
Answer:
B. Override update thus

```
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.

Question: 28
What is the effect of issuing a wait() method on an object

A. If a notify() method has already been sent to that object then it has no effect
B. The object issuing the call to wait() will halt until another object sends a notify() or notifyAll() method
C. An exception will be raised
D. The object issuing the call to wait() will be automatically synchronized with any other objects using the receiving object.
Question Help.
Answer:
B.

Question: 29
Which declarations for the main() method in a stand-alone program are valid?

A. public static void main()

B. public static void main(String[] string)

C. public static void main(String args)

D. static public int main(String[] args)

E. static void main(String[] args)
Question Help.
Answer:
B. is the only valid declaration (even though we've used the parameter name string rather than the usual args).

Answer:
A., C., D., and E. are not valid.

Question: 30
Which of the following methods are defined on the Graphics class:

A. drawLine(int, int, int, int)
B. drawImage(Image, int, int, ImageObserver)
C. drawString(String, int, int)
D. add(Component);
E. setVisible(boolean);
F. setLayout(Object);
Question Help.
Answer:
A., B. and C.

Question: 31
What will happen when you attempt to compile and run the following code?.

A. It will compile and the run method will print out the increasing value of i.
B. It will compile and calling start will print out the increasing value of i.
C. The code will cause an error at compile time.
D. 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
```

Question Help.

Answer:
C. The code will cause an error at compile time

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

Any class that is 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();

Question: 32
What will the following block of code write to the standard output when it is executed?

```
int i = 3;
int j = 0;
double k = 3.2;
if (i < k)
    if (i == j)
        System.out.println(i);
    else
        System.out.println(j);
else
    System.out.println(k);
```

A. 3

B. 0

C. 3.2

D. none of these

Question Help.

Answer:
B. is correct.

Question: 33
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
```

A. A value of 0 will be printed out
B. Nothing will be printed out
C. A compile time error
D. A compile time error complaining of the scope of the variable i
Question Help.
Answer:
C. A compile time error

Explanation:
Because only one instance of a static method exists no 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

Question: 34
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?

A. The string "Dolly" will be displayed at the centre of the frame
B. An error at compilation complaining at the signature of the paint method
C. The lower part of the word Dolly will be seen at the top of the form, with the top hidden.
D. The string "Dolly" will be shown at the bottom of the form
Question Help.
Answer:
C. The lower part of the word Dolly will be seen at the top of the form

Explanation:
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.

Question: 35

What is the name of the method used to schedule
a thread for execution?

A. init();
B. start();
C. run();
D. resume();
E. sleep();
Question Help.
Answer:
B.

Question: 36
In order to cause the paint(Graphics) method to execute,
which of the following is the most appropriate method
to call:

A. paint()
B. repaint()
C. paint(Graphics)
D. update(Graphics)
E. None - you should never cause paint(Graphics) to execute
Question Help.
Answer:
B.

Question: 37
Which of the following illustrates the correct way to pass a parameter into an applet:

A. <applet code=Test.class age=33 width=100 height=100>
B. <param name=age value=33>
C. <applet code=Test.class name=age value=33 width=100 height=100>
D. <applet Test 33>
Question Help.
Answer:
B.

Question: 38
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 .

A. You cannot be certain when garbage collection will run
B. Use the Runtime.gc() method to force garbage collection
C. Ensure that all the variables you require to be garbage
collected are set to null
D. Use the System.gc() method to force garbage collection
Question Help.
Answer:
A. You cannot be certain when garbage collection will run.

Explanation:
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 abou 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.

Question: 39
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));
```

A. Bic
B. ic
C. icy
D. error: no method matching substring(int,char)
Question Help.
Answer:
B. ic

Explanation:
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".

Question: 40
What is the effect of adding the sixth element to a vector created in the following manner:

```
new Vector(5, 10);
```

A. An IndexOutOfBounds exception is raised.
B. The vector grows in size to a capacity of 10 elements
C. The vector grows in size to a capacity of 15 elements
D. Nothing, the vector will have grown when the fifth element was added
Question Help.
Answer:
C.

Question: 41
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""");
}
```

Question Help.

Answer:

2. default, zero

Explanation:

Although it is normally placed last the default default statement does not have to be the last item as you fall through the case bock 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.

Question: 42

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
    }
}
```

A. On the line After //One put Base(10);
B. On the line After //One put super(10);
C. On the line After //Two put super(10);
D. On the line After //Three put super(10);

Question Help.

Answer:

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

Explanation:
Constructors can only be invoked from within constructors.

Question: 43
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
```

A. Compile Error: Methods in Base not found
B. Compile Error: Unable to access protected method in base class
C. Compilation followed by the output "amethod"
D. Compile error: Superclass Class1.Base of class Class1.Class1 not found
Question Help.
Answer:
D. Compile error: Superclass Class1.Base of class Class1.Class1 not found

Explanation:
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.

Question: 44
Which of the following statements about try, catch, and finally are true?

A. A try block must always be followed by a catch

block

B. A try block can be followed either by a catch block or a finally block, or both

C. A catch block must always be associated with a try block

D. A finally can never stand on its own (that is, without being associated with try block)

E. None of these are true
Question Help.
Answer:
B., C., and D.

Explanation:
A try block does not need to be followed by a catch block -- as long as it is followed by a finally block, so answer a is not correct.

Question: 45
Choose all true statements about the paint() method defined in the Component class:

A. it is protected

B. it takes an instance of class Graphics

C. it is static

D. it is invoked automatically whenever you minimize and then maximize a component, such as a window

E. there is also a version that takes an int
Question Help.
Answer:
B. and D. are the only correct answers.

Question: 46
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");
    }
}
```

A. if(s==s2)
B. if(s.equals(s2)
C. if(s.equalsIgnoreCase(s2))
D. if(s.noCaseMatch(s2))
Question Help.
Answer:
C. if(s.equalsIgnoreCase(s2))

Explanation:
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.

Question: 47
What happens when you try to compile and run the following program?

```
class Mystery {
    String s;
    public static void main(String[] args) {
        Mystery m = new Mystery();
        m.go();
    }
    void Mystery() {
        s = "constructor";
    }
    void go() {
        System.out.println(s);
    }
}
```

A. this code will not compile

B. this code compiles but throws an exception at runtime

C. this code runs but nothing appears in the standard output

D. this code runs and "constructor" in the standard output

E. this code runs and writes "null" in the standard output
Question Help.
Answer:
E.

Explanation:
Note that there is no real constructor: what appears to be a constructor is really a method, because it specifies a return type (void). A real constructor does not have a return type. Hence, s is never initialized, and so null appears in the standard output.

Question: 48
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);
    }
}
```

A. Error at compile time
B. 200
C. 100 followed by 200
D. 100
Question Help.
Answer:
A. 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.

Question: 49
How would you go about opening an image file called MyPicture.jpg?

A. Graphics.getGraphics("MyPicture.jpg");
B. Image image=Toolkit.getDefaultToolkit().getImage("MyPicture.jpg");
C. Graphics.openImage("MyPicture");
D. Image m=new Image("MyPicture");
Question Help.
Answer:
B. Image image=Toolkit.getDefaultToolkit().getImage("MyPicture.jpg");

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 B.

Question: 50
What does element zero of the string array
passed to the public static void main method contain?

A. The name of the program
B. The number of arguments
C. The first argument if one is present
Question Help.
Answer:
C.

Question: 51
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?

A. s3=s1 + s2;
B. s3=s1-s2;
C. s3=s1 & s2
D. s3=s1 && s2
Question Help.
Answer:
A. s3=s1 + s2;

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

Question: 52
What statements are true concerning the method
notify() that is used in conjunction with wait()?

A. if there is more than one thread waiting on a condition,
only the thread that has been waiting the longest is
notified

B. if there is more than one thread waiting on a condition,
there is no way to predict which thread will be notifed

C. notify() is defined in the Thread class

D. it is not strictly necessary to own the lock for the object
you invoke notify() for

E. notify() should only be invoked from within a while loop
Question Help.
Answer:

B. is the only correct answer.

Explanation:
A., C., and D. are definitely false.
For E., one common construct is to invoke wait()
in a while loop, but it is not necessary to invoke notify()
or wait() in a while loop.

Question: 53
What will the result be for the following block
of code when it is executed?

```
int i = 3;
int j = 0;
float k = 3.2F;
long m = -3;
if (Math.ceil(i) < Math.floor(k))
    if (Math.abs(i) == m)
        System.out.println(i);
    else
        System.out.println(j);
else
    System.out.println(Math.abs(m) + 1);
```

A. 3

B. 0

C. -3

D. 4

E. none of these
Question Help.
Answer:
D.

Explanation:
Math.floor(3.2) is 3, so the first if() expression yields
false, taking us to the final else clause.

Question: 54
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""");
```

```
        }
```
Question Help.
Answer:
3. one, two, default

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

Question: 55
Which of the following lines will compile without warning or error.
Question Help.
Answer:
int i=10;

Explanation:
Option:
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

Option:
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';

Option:
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

Option:
boolean b=null
Will not compile because a boolean value can either be true or false, null is not allowed.

Question: 56
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);
    }
}
```

A. Compile time error referring to a cast problem
B. A random number between 1 and 10
C. A random number between 0 and 1
D. A compile time error about random being an unrecognised method
Question Help.
Answer:
A. Compile time error referring to a cast problem

Explanantion:
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.

Question: 57
Which correctly create an array of five empty Strings?

A. String a [] = new String [5];
for (int i = 0; i 5; a[i++] = "");

B. String a [] = {"", "", "", "", ""};

C. String a [5];

D. String [5] a;

E. String [] a = new String [5];
for (int i = 0; i 5; a[i++] = null);
Question Help.
Answer:
A. and B.

Question: 58
Which of the following can you perform using the File class?

A. Change the current directory
B. Return the name of the parent directory
C. Delete a file
D. Find if a file contains text or binary information
Question Help.
Answer:
B. Return the name of the parent directory
C. 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.

Question: 59
What is the result of the following operation?

System.out.println(4 | 3);

A. 6
B. 0
C. 1
D. 7
Question Help.
Answer:
D. 7

Explanation:
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.

Question: 60
What is written to the standard output as the result of executing the following statements?

```
Boolean b1 = new Boolean(true);
Boolean b2 = new Boolean(true);
Object obj1 = (Object)b1;
Object obj2 = (Object)b2;

if (obj1 == obj2)
    if (obj1.equals(obj2))
        System.out.println("a");
    else
        System.out.println("b");
else
    if (obj1.equals(obj2))
        System.out.println("c");
    else
        System.out.println("d");
```

Select the one right answer.

A. a

B. b

C. c

D. d
Question Help.
Answer:
C.

Explanation:

Why? Because even though we've cast the Boolean objects to be of type Object, the method invoked still goes to the actual object type, which is Boolean. So, as is the case here, equals() will return true in the case of Boolean objects assigned to the same boolean value.

Question: 61
Which of the following methods can be legally inserted in place of the comment //Method Here ?

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

A. void amethod(int i) throws Exception {}
B. void amethod(long i)throws Exception {}
C. void amethod(long i){}
D. public void amethod(int i) throws Exception {}
Question Help.
Answer:
B. and C.

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

Question: 62
What is the legal range of a byte integral type?

A. 0 - 65,535
B. (-128) - 127
C. (-32,768) - 32,767
D. (-256) - 255
Question Help.
Answer:
C.

Question: 63
Analyze the following two classes.

```
class First {
    static int a = 3;
}

final class Second extends First {
```

```
    void method() {
        System.out.println(a);
    }
}
```

A. Class First compiles, but class Second does not

B. Class Second compiles, but class First does not

C. Neither class compiles

D. Both classes compile, and if method() is invoked,
it writes 3 to the standard output

E. Both classes compile, but if method() is invoked,
it throws an exception
Question Help.
Answer:
D.

Explanation:
This code is perfectly fine, and Second can access
the static variable in its superclass.

Question: 64
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]);
        }
}
```

A. error Can't make static reference to void amethod.
B. error method main not correct
C. error array must include parameter
D. amethod must be declared with String
Question Help.
Answer:
A. Can't make static reference to void amethod.

Explanation:
Because main is defined as static you need to create an instance
that the main exists in before you can call any of its methods.
Thus a typical way to do this would be:-

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

B. is an attempt to confuse because the convention is for a
main method arguments is 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 C. and D. are just nonsense.

Question: 65
What is wrong with the following code?

```
final class First {
    private int a = 1;
    int b = 2;
}

class Second extends First {
    public void method() {
        System.out.println(a + b);
    }
}
```

A. You cannot invoke println() without passing it a String

B. Since a is private, no classes other than First can access it

C. Second cannot extend First

D. final is not a valid keyword for a class
Question Help.
Answer:
B. and C.

Question: 66
Given the following class:

```
class Counter {
    public int startHere = 1;
    public int endHere = 100;
    public static void main(String[] args) {
        new Counter().go();
    }
    void go() {
        // A
        Thread t = new Thread(a);
        t.start();
    }
}
```

What block of code can you replace at line A above so that this program will count from startHere to endHere?

Select all valid answers.

A.

```
Runnable a = new Runnable() {
    public void run() {
        for (int i = startHere; i <= endHere; i++) {
            System.out.println(i);
        }
    }
};
```

B.

```
a implements Runnable {
    public void run() {
        for (int i = startHere; i <= endHere; i++) {
            System.out.println(i);
        }
   }
};
```

C.

```
Thread a = new Thread() {
    public void run() {
        for (int i = startHere; i <= endHere; i++) {
            System.out.println(i);
        }
    }
};
```

Question Help.

Answer:
A. and C.

Explanation:
B. is definitely not correct. A. and C. are anonymous inner classes that implement Runnable.

Question: 67
Which methods can be legally applied to a string object?

A. equals(String)
B. equals(Object)
C. trim()
D. round()
E. toString()

Question Help.

Answer:
A., B., C. and E.

Question: 68
What code placed after the comment:
// Start 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]);
       }
    }
}
```

A. for(int i=0; i < ia.length(); i++)
B. for (int i=0; i < ia.length(); i++)
C. for(int i=1; i < 4; i++)
D. for(int i=0; i < ia.length;i++)
Question Help.
Answer:
D. for(int i=0; i < ia.length;i++)

Explanation:
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.

Question: 69
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).

A. With a fixed font you will see 5 characters, with a proportional it will depend on the width of the characters
B. With a fixed font you will see 5 characters,with a proportional it will cause the field to expand to fit the text
C. The columns setting does not affect the number of characters displayed
D. Both will show exactly 5 characters
Question Help.
Answer:
A. With a fixed font you will see 5 characters, with a proportional it will depend on the width of the characters

Explanation:
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.

Question: 70
If the following HTML code is used to display the applet in the code MgAp what will be displayed at the console?

A. Error: no such parameter
B. 0
C. null
D. 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"));
        }
}
```

Question Help.
Answer:
C. null

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

Question: 71
What expressions are true concerning the following lines of code?

```
int[] arr = {1, 2, 3};
for (int i=0; i < 2; i++)
    arr[i] = 0;
```

A. arr[0] == 0

B. arr[0] == 1

C. arr[1] == 1

D. arr[2] == 0

E. arr[3] == 0

Question Help.
Answer:
A. is the only correct answer here.

Question: 72
Examine the following switch block:

```
char mychar = 'c';
switch (mychar) {
    default:
    case 'a': System.out.println("a"); break;
    case 'b': System.out.println("b"); break;
}
```

Which of the following questions are definitely true?

A. This switch block is illegal, because only integers
can be used in the switch statement.

B. This switch block is fine.

C. This switch block is illegal, because the default
statement must come last.

D. When this code runs, nothing is written to the standard
output.

E. When this code runs, the letter "a" is written to the
standard output.
Question Help.
Answer:
B. and E. are both correct.

Explanation:
It is perfectly valid to put the default case first. Since
there is no break statement at the end of the default,
the 'a' case is always run if there is no match for 'a' or 'b'.

Question: 73
How can you change the current working directory
using an instance of the File class called FileName?

A. FileName.chdir("DirName")
B. FileName.cd("DirName")
C. FileName.cwd("DirName")
D. The File class does not support directly changing the
current directory.
Question Help.
Answer:
D. The File class does not support directly changing the
current directory.

Explanation:
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.

Question: 74
For a variable width font, how "wide" is a TextField
created using the expression:

new TextField(20)

A. 20 times the average of all the characters in the
font used for this TextField object

B. 20 times the width of the letter M

C. 20 times the width of the letter a

D. 20 inches

E. 20 picas
Question Help.
Answer:
A.

Explanation:
Variable width fonts are based on the average of the letters when specifying the width.

Question: 75
Which of the following are correct?

A. 128 >> 1 gives 64
B. 128 >>> 1 gives 64
C. 128 >> 1 gives -64
D. 128 >>> 1 gives -64
Question Help.
Answer:
A. and B.

Question: 76
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

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

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

C.
```
public void ioCall (){
```

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

D. public void ioCall throws IOException(){
        DataInputStream din = new DataInputStream(System.in);
        din.readChar();
    }
```
Question Help.
Answer:
```
A. public void ioCall ()throws IOException {
         DataInputStream din = new DataInputStream(System.in);
         din.readChar();
    }
```

Explanation:
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.

Question: 77
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);
                        }
                    }
}
```
Question Help.
Answer:
Value for i=1 Value for j=1
Value for i=2 Value for j=1

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

Question: 78
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);
    }
}
```

A. 12
B. 012
C. 10
D. 10.0
Question Help.
Answer:
C. 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.

Question: 79
What does the following program do when it is run with the command:

java Mystery Mighty Mouse

```
class Mystery {
    public static void main(String[] args) {
        Changer c = new Changer();
        c.method(args);
        System.out.println(args[0] + " " + args[1]);
    }
    static class Changer {
        void method(String[] s) {
            String temp = s[0];
            s[0] = s[1];
            s[1] = temp;
        }
    }
}
```

A. This program causes an ArrayIndexOutOfBoundsException to be thrown

B. This program runs but does not write anything to the standard output

C. This program writes "Mighty Mouse" to the standard output

D. This program writes "Mouse Mighty" to the standard output
Question Help.

Answer:
D.

Explanation:
The method in the inner class changes the entries in the array, because objects are passed by reference (non-objects are passed by value).

Question: 80
What is the final value of temp in this sequence?

```
long temp = (int)3.9;
temp %= 2;
```

A. 0

B. 1

C. 2

D. 3

E. 4
Question Help.
Answer:
B.

Explanation:
temp is truncated to 3. The result of 3 % 2 is 1.

Question: 81
What tags are mandatory when creating HTML to display an applet?

A. name, height, width
B. code, name
C. codebase, height, width
D. code, height, width
Question Help.
Answer:
D. code, height, width

Question: 82
Which of the following will compile without error:-

A.

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

B.

```
package MyPackage;
import java.awt.*;
```

```
class MyClass{}
```

C.

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

Question Help.

Answer:
B. and C. will compile without error.

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

Question: 83
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;
    }
}
```

A. Error at compile time
B. An output of 99
C. An output of 198
D. An error at runtime

Question Help.

Answer:
A. Error at compile time

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

Question: 84
Analyze these two consequetive lines of code:

```
float f = 3.2;
int i = f;
```

A. this code would not compile

B. this code would compile and i would be set to 3

C. the second line could compile if it were written instead as:

```
int i = (byte)f;
```

D. the first line could compile if it were written instead as:

```
float f = 3.2F;
```

Question Help.

Answer:
A., C., and D.

Explanation:
The default value for a floating-point number is a double. Since a double is greater precision than a float, a floating-point constant assigned to a float data type must be cast or specified as a float by appending the number with an F.

Question: 85
Examine the following code which includes an inner class:

```
public final class Test4 implements A {

    class Inner {
        void test() {
            if (Test4.this.flag); {
                sample();
            }
        }
    }

    private boolean flag = false;

    public void sample() {
        System.out.println("Sample");
    }

    public Test4() {
        (new Inner()).test();
    }

    public static void main(String args []) {
        new Test4();
    }
}
```

What is the result:

A. Prints out "Sample"

B. Program produces no output but terminates correctly.

C. Program does not terminate.

D. The program will not compile

Question Help.
Answer:
A.

Question: 86
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]);
        }
}
```

A. Error: anar is referenced before it is initialized
B. null
C. 0
D. 5
Question Help.
Answer:
C. 0

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

Question: 87
What appears in the standard output if the method named problem() in the code below throws an instance of class Exception when the method named trythis() is invoked?

```
public void trythis() {
    try {
        System.out.println("1");
        problem();
    } catch (RuntimeException x) {
        System.out.println("2");
        return;
    } catch (Exception x) {
        System.out.println("3");
        return;
    } finally {
        System.out.println("4");
    }
    System.out.println("5");
}
```

A. "1"

B. "2"

C. "3"

D. "4"

E. "5"
Question Help.
Answer:
A., C., and D.

Explanation:
Note that even with a return statement at the end of
the catch block, the finally block is still executed.

Question: 88
What will happen when you compile the following code?

```
public class MyClass{
        static int i;

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

A. Error Variable i may not have been initialized
B. null
C. 1
D. 0
Question Help.
Answer:
D. 0

Explanation:
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 :-

A. Error Variable i may not have been initialized

Question: 89
A byte can be of what size

A. -128 to 127
B. (-2 power 8)-1 to 2 power 8
C. -255 to 256
D. depends on the particular implementation of the Java
Virtual machine
Question Help.
Answer:
A. byte is a signed 8 bit integer.

Question: 90
Which of the following will output -4.0?

A. System.out.println(Math.floor(-4.7));
B. System.out.println(Math.round(-4.7));

C. System.out.println(Math.ceil(-4.7));
D. System.out.println(Math.Min(-4.7));
Question Help.
Answer:
C. System.out.println(Math.ceil(-4.7));

Explantion:
Options A. and B. will produce -5 and option D.
will not compile because the Min method requires 2 parameters.

Question: 91
What is written to the standard output as the result of executing the following statements?

```
Boolean b1 = new Boolean(true);
Boolean b2 = new Boolean(true);

if (b1 == b2)
      if (b1.equals(b2))
           System.out.println("a");
      else
           System.out.println("b");
else
    if (b1.equals(b2))
         System.out.println("c");
    else
         System.out.println("d");
```

A. a

B. b

C. c

D. d
Question Help.
Answer:
C.

Explanation:
b1 and b2 are different objects but the Boolean class overrides the equals() test to return true if the two objects being compared have the same boolean value.

Question: 92
Which of the following lines of code will compile without error?

A.

```
int i=0;
if(i)
            System.out.println("""Hello""");
            }
```

B.

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

C.

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

D.

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

Question Help.
Answer:
B. and C.

A. 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.

Question: 93
What order can you place the following pieces of
a source file in so that the source file will compile
without errors or warnings?

```
//A
import java.applet.*;

//B
class Helper {
}

//C
package myclasses;

//D
public class MyApplet extends java.applet.Applet {
}
```

a) A, B, C, D

b) A, C, B, D

c) C, A, B, D

d) C, A, D, B

e) C, B, A, D

Question Help.
Answer:

c and d are correct.

Explanation:
The package statement must come first, and import second. After that, classes can be declared in any order. (There can only be one public class per source file.)

Question: 94
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?

A. mid(2,s);
B. charAt(2);
C. s.indexOf('v');
D. indexOf(s,'v');
Question Help.
Answer:
C. s.indexOf('v');

Explanation:
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.

Question: 95
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)
        }
            System.out.println("Doing finally");
        }
       return 0;
    }
}
```

A. No such file found
B. No such file found ,-1
C. No such file found, doing finally, -1

D. 0
Question Help.
Answer:
C. No such file found, doing finally, -1

Explanation:
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.

Question: 96
Why won't the following class compile?

```
class A {
    private int x;
    public static void main(String[] args) {
        new B();
    }
    class B {
        B() {
            System.out.println(x);
        }
    }
}
```

A. Class B tries to access a private variable defined in its ouer class.

B. Class A attempts to create an instance of B when there is no current instance of class A.

C. Class B's constructor must be public.
Question Help.
Answer:
B.

Explanation:
This could be made to compile by making class B static, or by creating class B from an instance method.

Question: 97
You have an 8-bit file using the character set defined by ISO 8859-8. You are writing an application to display this file in a TextArea. The local encoding is already set to 8859-8. How can you write a chunk of code to read the first line from this file?

You have three variables accessible to you:

- myfile is the name of the file you want to read
- stream is an InputStream object associated with this file
- s is a String object

Select all valid answers.

A.

```
InputStreamReader reader = new InputStreamReader(stream, "8859-8");
BufferedReader buffer = new BufferedReader(reader);
s = buffer.readLine();
```

B.

```
InputStreamReader reader = new InputStreamReader(stream);
BufferedReader buffer = new BufferedReader(reader);
s = buffer.readLine();
```

C.

```
InputStreamReader reader = new InputStreamReader(myfile, "8859-8");
BufferedReader buffer = new BufferedReader(reader);
s = buffer.readLine();
```

D.

```
InputStreamReader reader = new InputStreamReader(myfile);
BufferedReader buffer = new BufferedReader(reader);
s = buffer.readLine();
```

E.

```
FileReader reader = new FileReader(myfile);
BufferedReader buffer = new BufferedReader(reader);
s = buffer.readLine();
```

Question Help.

Answer:
A., B., and E.

Explanation:
The other answers won't compile.

Question: 98
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?

A. s.setBackground(Color.pink);
B. s.setColor(PINK);
C. s.Background(pink);
D. s.color=Color.pink;

Question Help.

Answer:
A. s.setBackground(Color.pink);

Explanation:
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.

Question: 99
If g is a graphics instance what will the following code draw on the screen?.

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

A. 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.

B. 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.

C. 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.

D. 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.
Question Help.
Answer:
C. 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.

Explanation:
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.

Question: 100
The method setBackground() defined for the Graphics class:

A. takes an integer value

B. takes an instance of class Color

C. takes an instance of a Component subclass

D. sets the drawing mode for the associated Component object

E. sets the drawing color for the associated Component object

F. changes the background color for the associated Component object
Question Help.
Answer:

B. and F.

Question: 101
What will happen if you attempt to compile and run the following code?

A. Compile and run without error
B. Compile time Exception
C. 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;
        }

}
```

Question Help.
Answer:
C. Runtime Exception

Explanation:
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).

Question: 102
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])
    }
}
```

A. myprog
B. good
C. morning
D. Exception raised: "java.lang.ArrayIndexOutOfBoundsException: 2"

Question Help.
Answer:
D.

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

Explantion:
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.

Question: 103
Which code fragments would correctly identify the number of arguments passed via the command line to a Java[tm] application, excluding the name of the class that is being invoked?

A. int count = args.length;

B. int count = args.length - 1;

C. int count = 0;
```
while (args[count] != null)
    count ++;
```

D. int count=0;
```
while (!(args[count].equals("")))
    count ++;
```
Question Help.
Answer:
A.

Question: 104
```
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?

A. public
B. private
C. static
D. friend
Question Help.
Answer:
A.,B. and C.

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

Question: 105
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]);
        }
}
```

A. a sequence of 5 0's will be printed
B. Error: ar is used before it is initialized
C. Error Mine must be declared abstract
D. IndexOutOfBoundes Error
Question Help.
Answer:
C. Error Mine must be declared abstract

Explanation:
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.

Question: 106
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?

A. Define your own Listener interface according to the event to be tracked
B. Use the search facility in the HTML documentation for the listener needed
C. Move up the hierarchy in the HTML documentation to locate methods in base classes
D. Subclass awt.event with the appropriate Listener method
Question Help.
Answer:

C. Move up the hierarchy in the HTML documentation to locate methods in base classes

Explanation:
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.

Question: 107
What will happen if you try to compile and execute B's main() method?

```
class A {
    int i;
    A(int i) {
        this.i = i * 2;
    }
}

class B extends A {
    public static void main(String[] args) {
        B b = new B(2);
    }
    B(int i) {
        System.out.println(i);
    }
}
```

A. The instance variable i is set to 4

B. The instance variable i is set to 2

C. The instance variable i is set to 0

D. This code will not compile
Question Help.
Answer:
D.

Explanation:
Because B's constructor implicitly calls A's no-args constructor, and such a constructor is not defined.

Question: 108
Analyze the following code.

```
void looper() {
    int x = 0;
one:
    while (x < 10) {
two:
        System.out.println(++x);
        if (x > 3)
            break two;
    }
}
```

A. This code compiles

B. This code does not compile

C. This method writes the number 0 to the standard output

D. the numbers 1 and 2 to the standard output

E. the number 3 to the standard output

F. the number 4 to the standard output

G. the numbers 5 through 9 to the standard output

H. the number 10 to the standard output
Question Help.
Answer:
B.

Explanation:
This code does not compile! Because the label two: is not associated with a loop.

Question: 109
Analyze the following code:

```
class WhatHappens implements Runnable {
    public static void main(String[] args) {
        Thread t = new Thread(this);
        t.start();
    }
    public void run() {
        System.out.println("hi");
    }
}
```

A. This program does not compile

B. This program compiles but nothing appears in the standard output

C. This program compiles and the word "hi" appears in the standard output, once

D. This program compiles and the word "hi" appears continuously in the standard output until the user hits control-c to stop the program
Question Help.
Answer:
A.

Explanation:
This code does not compile because there is no this variable in the static main() method to pass to the Thread's constructor.

Question: 110
Which of the following identifiers are LEGAL?

A. #_pound

B. _underscore

C. 5Interstate

D. Interstate5

E. _5_

Question Help.
Answer:
B., D. and E.

Explanation:
A. and C. are illegal. Note that E. is legal: a variable can start with an underscore. Therefore, a variable does not have to contain any letters at all.

Question: 111
Which are correct class declarations? Assume in each case that the text constitutes the entire contents of a file called Fred.java on a system with a case-significant file system.

A.
```
public class Fred {
    public int x = 0;
    public Fred (int x) {
        this.x = x;
    }
}
```

B.
```
public class fred
    public int x = 0;
    public fred (int x) {
        this.x = x;
    }
}
```

C.
```
public class Fred extends MyBaseClass, MyOtherBaseClass {
    public int x = 0;
    public Fred (int xval) {
        x = xval;
    }
}
```

D.
```
protected class Fred {
    private int x = 0;
    private Fred (int xval) {
        x = xval;
    }
}
```

E.
```
import java.awt.*;
public class Fred extends Object {
    int x;
    private Fred (int xval) {
        x = xval;
    }
}
```
Question Help.
Answer:
A. and E.

Question: 112
Which statements accurately describe the following line

of code?

String[][] s = new String[10][];

A. This line of code is illegal.

B. s is a two-dimensional array containing 10 rows
and 10 columns

C. s is an array of 10 arrays.

D. Each element in s is set to ""

E. Each element in s is uninitialized and must be
initialized before it is referenced.
Question Help.
Answer:
C. is the only correct answer.

Explanation:
The entries in an array of objects all are set to null
if no explicit value is given.

Question: 112
Which statements accurately describe the following line
of code?

String[][] s = new String[10][];

A. This line of code is illegal.

B. s is a two-dimensional array containing 10 rows
and 10 columns

C. s is an array of 10 arrays.

D. Each element in s is set to ""

E. Each element in s is uninitialized and must be
initialized before it is referenced.
Question Help.
Answer:
C. is the only correct answer.

Explanation:
The entries in an array of objects all are set to null
if no explicit value is given.

Question: 114
Given this interface definition:

```
interface A {
    int method1(int i);
    int method2(int j);
}
```

which of the following classes implement this interface and is not abstract?

A.
```
class B implements A {
    int method1() { }
    int method2() { }
}
```

B.
```
class B {
    int method1(int i) { }
    int method2(int j) { }
}
```

C.
```
class B implements A {
    int method1(int i) { }
    int method2(int j) { }
}
```

D.
```
class B extends A {
    int method1(int i) { }
    int method2(int j) { }
}
```

E.
```
class B implements A
    int method2(int j) { }
    int method1(int i) { }
}
```
Question Help.
Answer:
C. and E. are the only correct answers.

Explanation:
B. is a concrete class all right, but it does not implement the interface.

Question: 115
How can you write a line of code for an applet's init() method that determines how wide the applet is?

A. int width = this.getY();
B. int width = this.getSize().w;
C. int width = getSize();
D. int width = getSize().w;
E. int width = getWidth();
Question Help.
Answer:
B. and D.

Explanation:
The other answers are either partial answers or won't compile.

Question: 116
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");
    }
}
```

A. Compile time error complaining that Base.amethod is private
B. Runtime error complaining that Base.amethod is private
C. Output of Base.amethod
D. Output of Over.amethod()
Question Help.
Answer:
D. Output of Over.amethod()

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

Question: 117
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]);
        }
}
```

A. 1
B. Error anar is referenced before it is initialized
C. 2
D. Error: size of array must be defined
Question Help.
Answer:
C. 2

Explanation:
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.

Question: 118
Given the following class definition:

```
class A {
    protected int i;
    A(int i) {
        this.i = i;
    }
}
```

Which of the following would be a valid inner class for this class?

Select all valid answers.

A.

```
class B {
}
```

B.

```
class B extends A {
}
```

C.

```
class B {
    B()
        System.out.println("i = " + i);
    }
}
```

D.

```
class B {
    class A {
    }
}
```

E.

```
class A {
}
```

Question Help.
Answer:
A. is the only right answer.

Explanation:
For B. and C., B extends A but implicitly invokes the no-args constructor, which A does not define. For D. and E., an inner class cannot be the same name as any of its enclosing classes.

Question: 119
Given the following class definition:

```
class A {
    public int x;
    private int y;
    class B {
        protected void method1() {
        }
        class C {
            private void method2() {
            }
        }
    }
}

class D extends A {
    public float z;
}
```

What can method2() access directly, without a reference to another instance?

A. the variable x defined in A

B. the variable y defined in A

C. method1 defined in B

D. the variable z defined in D

Question Help.

Answer:
A., B., and C.

Explanation:
D. is not correct -- an inner class cannot access a variable in a subclass of an enclosing class without a reference to it.

Question: 120
Describe the following applet.

```
import java.applet.Applet;
import java.awt.event.*;
import java.awt.*;
public class MyApplet extends Applet {
    Button b1, b2;
    public void init() {
        ActionListener a = new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                if (evt.getSource() == b1) {
                    b1.setEnabled(false);
                    b2.setEnabled(true);
                } else {
                    b1.setEnabled(true);
```

```
                    b2.setEnabled(false);
                }
            }
        };
        b1 = new Button("1");
        b1.addActionListener(a);
        add(b1);
        b2 = new Button("2");
        b2.addActionListener(a);
        add(b2);
    }
}
```

A. Nothing appears in the applet

B. One button appears in the applet but does nothing

C. Two buttons appear in the applet

D. When the user clicks a button, nothing happens

E. When the user clicks a button, it becomes disabled

F. When a user clicks a button, the other button becomes enabled
Question Help.
Answer:
C., E., and F. all describe this
applet accurately.

Question: 121
Given the following class:

```
class Counter {
    public static void main(String[] args) {
        Thread t = new Thread(new CounterBehavior());
        t.start();
    }
}
```

Which of the following is a valid definition of
CounterBehavior that would make Counter's main()
method count from 1 to 100, counting once per second?

A. This class is an inner class to Counter:

```
class CounterBehavior {
    for (int i = 1; i <= 100; i++);
        try {
            System.out.println(i);
            Thread.sleep(1000);
        } catch (InterruptedException x) {}
    }
}
```

B. This class is an inner class to Counter:

```
class CounterBehavior implements Runnable {
    public void run() {
```

```
        for (int i = 1; i <= 100; i++);
            try {
                System.out.println(i);
                Thread.sleep(1000);
            } catch (InterruptedException x) {}
        }
    }
}
```

C. This class is a top-level class:

```
static class CounterBehavior implements Runnable {
    public void run() {
        try {
            for (int i = 1; i <= 100; i++) {
                System.out.println(i);
                Thread.sleep(1000);
            }
        } catch (InterruptedException x) {}
    }
}
```

Question Help.

Answer:

C. is the only correct answer.

Explanation:

B. is not correct because the inner class is needed in main(), where there is no instance of the enclosing class (Counter). Only a top-level class will work in this case.

Question: 122

Which cannot be added to a Container?

A. an Applet

B. a Component

C. a Container

D. a Menu

E. a Panel

Question Help.

Answer:

D.