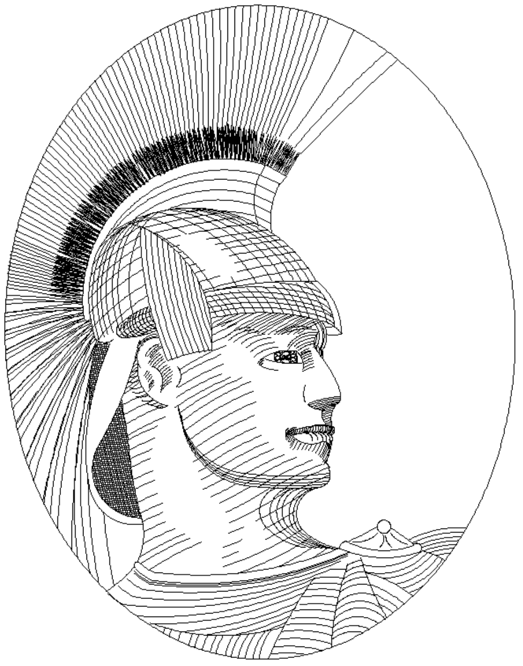
Cram Bible Lab



EasyJAVA Cram Bible

Sun Certified Java Programmer

Exam 310-022/310-025

(Release 1.0)

(Part 2) SUN CERTIFIED PROGRAMMER FOR THE JAVA 2 PLATFORM
(130 Questions and Explanations)

Question: 1
Which of the following statements are true?

1) The String class is implemented as a char array, elements
are addressed using the stringname[] convention

2) Strings are implemented as a class for which Java
overloads the + operator

3) Strings are a primitive type in Java and the StringBuffer
is used as the matching wrapper type

4) The size of a string can be retrieved using the length
property
Question Help.
2) Strings are implemented as a class for which Java
overloads the + operator

In Java Strings are implemented as a classs within
the Java.lang package with the special distinction
that the + operator is overloaded.

If you thought that the String class is implemented
as a char array, you may have a head full of C/++
that needs emptying of legacy information and
upsizing to the new paradigm. There is no
"wrapper class" for String as wrappers are only
for primitive types.

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

Question: 2
Which of the following statements are true?

1) A class defined within a method can only access
static methods of the enclosing method.

2) A class defined within a method can only access final
methods of the enclosing method.

3) A class defined with a method cannot access any of
the fields of the enclosing method.

4) A class defined within a method can access any fields
accessible by the enclosing method.
Question Help.
A class defined within a method can only access final
methods of the enclosing method

Such a class can access parameters passed to the enclosing method

Question: 3
Given the following variables

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

Which of the following will compile without error?

A. c=c+i;
B. s+=i;
C. i+=s;
D. c+=s;
  Question Help.
Answer:
B. s+=i;

Explanation:
Only a String acts as if the + operator were overloaded

Question: 4
What will be output by the following line?

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

A. -2
B. 2.0
C. -3
D. -3.0
  Question Help.
Answer:
D. -3.0

Question: 5
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: 6
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: 7
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: 8
Which of the following statements are true?

A. At the root of the collection hierarchy is a
   class called Collection
B. The collection interface contains a method
   called enumerator
C. The interator method returns an instance
   of the Vector class
D. The set interface is designed for unique
   elements
   Question Help.

Answer:
D. The set is designed for unique elements.

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

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

```
//File P1.java

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

//File P2.java

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

A. Both compile and P2 outputs "What a fancy method"
   when run
B. Neither will compile
C. Both compile but P2 has an error at run time
D. P1 compiles cleanly but P2 has an error at compile

time
Question Help.
Answer:
D. P1 compiles cleanly but P2 has an error at compile
time

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

Question: 10
Question 9)

Which of the following are methods of the Runnable interface?

1) run
2) start
3) yield
4) stop
Question Help.
Although you start a Thread running with the start
method it is not part of the Runnable interface.
The Runnable interface only has the one method
called run

Question: 11
When using the GridBagLayout manager, each new
component requires a new instance of the
GridBagConstraints class. Is this statement

A. true
B. false
Question Help.
Answer:
B. false

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

Question: 12
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: 13
What can cause a thread to stop executing?

A. The program exits via a call to exit(0);
B. The priority of another thread is increased
C. A call to the stop method of the Thread class
D. A call to the halt method of the Thread class
Question Help.
Answer:
A. The program exits via a call to exit(0);
B. The priority of another thread is increased
C. A call to the stop method of the Thread class

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

Question: 14
Which of the following are legal statements?

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

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

Question: 15
Given the following variables which of the following lines will compile without error?

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

A. j= i<<s;
B. j= i<<j;
C. j= i<<d;
D. j= i<<l;
   Question Help.
Answer:
B. j= i<<j;
D. j= i<<l;

Question: 16
Which of the following are fields of the GridBagConstraints class?

A. ipadx
B. fill
C. insets
D. width
   Question Help.
Answer:
A. ipadx
B. fill
C. insets

Question: 17
Which is the advantage of encapsulation?

A. Only public methods are needed.
B. No exceptions need to be thrown from any method.
C. Making the class final causes no consequential
   changes to other code.
D. It changes the implementation without changing the
   interface and causes no consequential changes to
   other code.
E. It changes the interface without changing the
   implementation and causes no consequential changes
   to other code.
   Question Help.

Answer:
D. It changes the implementation without changing the
interface and causes no consequential changes to
other code.

Question: 18
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: 19
What will be output by the following line?

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

1) -2
2) 2.0

3) -3
4) -3.0
   Question Help.
 4) -3.0

According to the JDK documentation the Math.floor method returns "the largest (closest to positive infinity) double value that is not greater than the argument and is equal to a mathematical integer."

Question: 20
What will happen when you attempt to compile and run this program?

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

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

}
```

A. Compile and run with output of "Outer"
B. Compile and run with output of "Inner"
C. Compile time error because Inner is declared as private
D. Compile time error because of the line creating the instance of Inner
   Question Help.
Answer:
D. Compile time error because of the line creating the instance of Inner

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

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

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

Question: 21
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
   Question Help.
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: 22
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: 23
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: 24
Which of the following is the correct syntax for suggesting that the JVM performs garbage collection?

A. System.free();
B. System.setGarbageCollection();
C. System.out.gc();
D. System.gc();
   Question Help.
Answer:
D. System.gc();

Question: 25
Which of the following are true ?

1) The elements of a Collection class can be ordered by using the
   sort method of the Collection interface.

2) You can create an ordered Collection by instantiating a class that
   implements the List interface.

3) The Collection interface sort method takes parameters of A or D
   to change the sort order, Ascending/Descending.

4) The elements of a Collection class can be ordered by using the order
   method of the Collection interface.
   Question Help.
2) You can create an ordered Collection by instantiating a class that
   implements the List interface

Question: 26
Which of the following statements about threading
are true?

A. You can only obtain a mutually exclusive lock on
   methods in a class that extends Thread or implements
   runnable
B. You can obtain a mutually exclusive lock on any object
C. A thread can obtain a mutex lock on a method declared
   with the keyword synchronized
D. Thread scheduling algorithms are platform dependent
   Question Help.
Answer:
B. You can obtain a mutually exclusive lock on any
   object
C. A thread can obtain a mutex lock on a method
   declared with the keyword synchronized
D. Thread scheduling algorithms are platform dependent

Explanation:
Yes that says dependent and not independent.

Question: 27
Which of the following are valid statements?

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

The Math class is final and cannot have sub-classes (children).
The max (and min) methods take two parameters not one. As you
might guess the round method takes one parameter, the number
that is to be rounded. There is no mod method in the Math class.

Question: 28
How can you implement encapsulation in a class ?

1) Make all variables protected and only allow access
   via methods

2) Make all variables private and only allow access via methods

3) Ensure all variables are represented by wrapper classes

4) Ensure all variables are accessed through methods in an
   ancestor class
   Question Help.
2) make all variables private and only allow access via methods

Question: 29
Which of the following is successfully create an
instance of the Vector class and add an element?

A. Vector v=new Vector(99);
   v[1]=99;

B. Vector v=new Vector();
   v.addElement(99);

C. Vector v=new Vector();
   v.add(99);

D. Vector v=new Vector(100);
   v.addElement("99");
   Question Help.
Answer:
D. Vector v=new Vector(100);
  v.addElement("99")

Explanantion:
A vector can only store objects not primitives. The parameter
"99" for the addElement method pases a string object to the
Vector.

Option A. creates a vector OK but then uses array syntax
to attempt to assign a primitive. Option B. also creates a
vector then uses correct Vector syntax but falls over when the
parameter is a primitive instead of an object. Option C.
compounds the errors by using the fictitious add method.

Question: 30
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: 31
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: 32
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: 33
What will happen when you attempt to compile
and run the following code?

```
public class Bground extends Thread{

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

A. A compile time error indicating that no run
   method is defined for the Thread class
B. A run time error indicating that no run method
   is defined for the Thread class
C. Clean compile and at run time the values 0 to 9
   are printed out
D. Clean compile but no output at runtime
   Question Help.
Answer:
D. Clean compile but no output at runtime

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

public void run()

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

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

Question: 34
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: 35
What will happen when you attempt to compile and run this code?

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

	}
}
```

A. The code will compile and run, printing out the words "My Func"
B. The compiler will complain that the Base class is not declared as abstract.
C. The code will compile but complain at run time that the Base class has non abstract methods
D. The compiler will complain that the method myfunc in the base class has no body, nobody at all to looove it.
Question Help.
Answer:
B. The compiler will complain that the Base class is not declared as abstract.

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

Question: 36
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?

A. The GridBagLayout manager does not allow you to do this
B. The setFill method of the GridBagLayout class
C. The setFill method of the GridBagConstraints class

D. The fill field of the GridBagConstraints class
  Question Help.
Answer:
D. The fill field of the GridBagConstraints class

Explanation:
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.

Question: 37
To use the Math class you must have the statement

import java.lang.Math;

in your file. Is this statement true or false?

  Question Help.
The Math class is part of the core lang package and does not need to be imported

Question: 38
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: 39
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: 40
Why might you define a method as native?

A. To get to access hardware that Java does not
   know about
B. To define a new data type such as an unsigned
   integer
C. To write optimised code for performance in a language
    such as C/C++
D. To overcome the limitation of the private scope of a
   method
   Question Help.
Answer:
A. To get to access hardware that Java does not know
   about
C. To write optimised code for performance in a language
   such as C/C++

Question: 41

```
   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: 42
What will happen when you attempt to compile and
run the following code with the command line
"hello there"?

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

A. Compile time error
B. Compilation and output of "hello"
C. Compilation and output of "there"

D. None of the above
  Question Help.
Answer:
A. Compile time error

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

Question: 43
Under what circumstances might you use the yield method of the Thread class

A. To call from the currently running thread to allow another thread of the same priority to run
B. To call on a waiting thread to allow it to run
C. To allow a thread of higher priority to run
D. To call from the currently running thread with a parameter designating which thread should be allowed to run.
  Question Help.
Answer:
A. To call from the currently running thread to allow another thread of the same priority to run

Question: 44
Which declares an abstract method in an abstract Java class?

A. public abstract method();
B. public abstract void method();
C. public void abstract Method();
D. public void method() {abstract;/}
E. public abstract void method() {/}
  Question Help.
Answer:
B. public abstract void method();

Question: 45
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: 46
Which statements are correct about the anchor field?

A. It is a field of the GridBagLayout manager for
   controlling component placement
B. It is a field of the GridBagConstraints class for
   controlling component placement
C. A valid setting for the anchor field is
   GridBagConstraints.NORTH
D. The anchor field controls the height of
   components added to a container
   Question Help.

Answer:
B. It is a field of the GridBagConstraints class for
   controlling component placement
C. A valid settting for the anchor field is
   GridBagconstraints.NORTH

Question: 47
What will happen when you attempt to compile
and run this code?

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

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

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

A. Error at compile time
B. Visible Frame created that that can be closed
C. Compilation but no output at run time
D. Error at compile time because of comment
   before import statements
   Question Help.
Answer:
A. Error at compile time

Explanation:
If you implement an interface you must create bodies for all methods in that interface. This code will produce an error saying that MyWc must be declared abstract because it does not define all of the methods in WindowListener. Option 1 is nonsense as comments can appear anywhere. Option 3 suggesting that it might compile but not produce output is ment to mislead on the basis that what looks like a constructor is actually an ordinary method as it has a return type.

Question: 48
How do you indicate where a component will be positioned using Flowlayout?

A. North, South,East,West
B. Assign a row/column grid reference
C. Pass a X/Y percentage parameter to the add
   method
D. Do nothing, the FlowLayout will position the
   component
   Question Help.
Answer:
D. Do nothing, the FlowLayout will position the component

Question: 49
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: 50
What will happen when you attempt to compile and run the following code?

```
import java.io.*;

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

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

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

}
```

A. Compile time error caused by protected constructor
B. Compile time error caused by amethod not declaring
   Exception
C. Runtime error caused by amethod not declaring
   Exception
D. Compile and run with output of "Pausing" and
   "Continuing" after a key is hit
   Question Help.
Answer:
D. Compile and run with output of "Pausing" and
   "Continuing" after a key is hit

Explanation:
An overriden method in a sub class must not throw

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

Question: 51
What will happen when you attempt to compile and run the following code?

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

A. Compilation error: Constructors cannot be declared protected
B. Run time error: Constructors cannot be declared protected
C. Compilation and running with output 0 to 10
D. Compilation and running with output 0 to 9
Question Help.
Answer:
D. Compilation and running with output 0 to 9

Question: 52
What best describes the apprearance of an applet with the following code?

```
import java.awt.*;

public class FlowAp extends Frame{

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

A. A Frame with buttons marked One to Four placed
   on each edge.
B. A Frame with buutons Makred One to four running
   from the top to bottom
C. A Frame with one large button marked Four in the
   Centre
D. An Error at run time indicating you have not set a
   LayoutManager
   Question Help.
Answer:
C. A Frame with one large button marked Four in the Centre

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

Question: 53
What will happen when you attempt to compile and run the following code

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

A. Compile error, attempting to peform binary
   comparison on logical data type
B. Compilation and output of "We are equal 10"
C. Compilation and output of "Not equal! 20"
D. Compilation and output of "Not equal! 10"
   Question Help.
Answer:
D. Compilation and output of "Not equal! 10"

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

Question: 54

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: 55
What will happen when you attempt to compile and run this code?

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

A. The compiler will complain that main is a
   reserved word and cannot be used for a class
B. The code will compile and when run will
   print out "Hello cruel world"
C. The code will compile but will complain at run
   time that no constructor is defined
D. The code will compile but will complain at run
   time that main is not correctly defined
   Question Help.
Answer:
D. The code will compile but will complain at run
   time that main is not correctly defined

Question: 56
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: 57
Which of the following are true?

1) Java uses a time-slicing scheduling system for determining
  which Thread will execute.

2) Java uses a pre-emptive, co-operative system for determining
  which Thread will execute.

3) Java scheduling is platform dependent and may vary from one
   implementation to another.

4) You can set the priority of a Thread in code.
  Question Help.
3) Unlike most aspects of Java, scheduling is platform dependent and
   may vary from one implementation to another.

4) You can set the priority of a Thread in code though the results will
  not be consistent on different platforms.

Question: 58
What will happen when you attempt to compile
and run the following code

```
public class MySwitch{

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

public void amethod(){

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

A. None of these options
B. Compile time errror target of switch must be an integral type
C. Compile and run with output "This is the default output"
D. Compile and run with output "ten"
   Question Help.
Answer:
A. None of these options

Explanation:
Because of the lack of a break statement after the break 10; statement the actual output will be "ten" followed by "twenty"

Question: 59
What most closely matches the appearance when this code runs?

```
import java.awt.*;

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

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

A. The buttons will run from left to right along the bottom of the Frame
B. The buttons will run from left to right along the top of the frame
C. The buttons will not be displayed
D. Only button three will show occupying all of the frame
   Question Help.
Answer:
B. The buttons will run from left to right along the top of the frame

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

Question: 60
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: 61
You want to loop through an array and stop when you come to the last element. Being a good java programmer and forgetting everything you ever knew about C/C++ you know that arrays contain information about their size. Which of the following can you use?

A. myarray.length();
B. myarray.length;
C. myarray.size
D. myarray.size();
Question Help.
Answer:
B. myarray.length;

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

Question: 62
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?

A. The GridBagLayout manager does not allow you to do this
B. The setFill method of the GridBagLayout class
C. The setFill method of the GridBagConstraints class
D. The fill field of the GridBagConstraints class
  Question Help.
Answer:
D. The fill field of the GridBagConstraints class

Explanation:
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.

Question: 63
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: 64
Which of the following are Java key words?

A. double
B. Switch
C. then
D. instanceof
Question Help.
Answer:
A. double
D. instanceof

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

Question: 65
What will happen when you attempt to compile and run the following code?

```
public class StrEq{

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

A. Compile time error caused by private constructor
B. Output of "we have a match"
C. Output of "Not equal"
D. Compile time error by attempting to compare
   strings using ==
   Question Help.
Answer:
C. Output of "Not equal"

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

Question: 66
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: 67
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: 68
What would be the result of attempting to compile
and run the following piece of code?

```
public class Test
   static int x;
   public static void main (String args[])
     System.out.println("Value is " + x);
   }
}
```

A. The output "Value is 0" is printed.
B. An object of type NullPointerException is thrown.
C. An "illegal array declaration syntax" compiler error
   occurs.
D. A "possible reference before assignment" compiler
   error occurs.
E.An object of type ArrayIndexOutOfBoundsException is
  thrown.
  Question Help.
Answer:
A.The output "Value is 0" is printed.

Question: 69
Which of the following will compile correctly?

A. short myshort = 99S;
B. String name = 'Excellent tutorial Mr Green';
C. char c = 17c;
D. int z = 015;

Question Help.
Answer:
D. int z = 015;

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

Question: 70
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: 71
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: 72
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: 73
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: 74
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: 75
What should you use to position a Button within an
application Frame so that the size of the Button is NOT
affected by the Frame size?

A. a FlowLayout
B. a GridLayout
C. the center area of a BorderLayout
D. the East or West area of a BorderLayout
E. the North or South area of a BorderLayout
  Question Help.
Answer:
A. a FlowLayout

Question: 76
Which of the following are benefits of encapsulation?

1) All variables can be manipulated as Objects instead
  of primitives

2) By making all variables protected they are protected from
  accidental corruption

3) The implementation of a class can be changed without
  breaking code that uses it

4) Making all methods protected prevents accidental corruption
  of data
  Question Help.
3) The implementation of a class can be changed
  without breaking code that uses it

Question: 77
Given the folowing classes which of the following
will compile without error?

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

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

A. o1=o2;
B. b=ob;
C. ob=b;
D. o1=b;
   Question Help.
Answer:
A. o1=o2;
B. b=ob;
D. o1=b;

Question: 78
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
   Question Help.
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: 79
What will happen when you attempt to compile and
run this code?

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

A. Compile time error: Base cannot be private
B. Compile time error indicating that an integer
   cannot be transient
C. Compile time error transient not a data type
D. Compile time error malformed main method
   Question Help.
Answer:
A. Compile time error: Base cannot be private

Explanation:
A top level (non nested) class cannot be private.

Question: 80
Your chief Software designer has shown you a sketch of the
new Computer parts system she is about to create. At the
top of the hierarchy is a Class called Computer and under
this are two child classes. One is called LinuxPC and one is
called WindowsPC. The main difference between the two is
that one runs the Linux operating System and the other runs
the Windows System (of course another difference is that one
needs constant re-booting and the other runs reliably). Under
the WindowsPC are two Sub classes one called Server and one
Called Workstation. How might you appraise your designers
work?

A. Give the goahead for further design using the current
   scheme
B. Ask for a re-design of the hierarchy with changing the
   Operating System to a field rather than Class type
C. Ask for the option of WindowsPC to be removed as it
   will soon be obsolete
D. Change the hierarchy to remove the need for the superfluous

  Computer Class.
  Question Help.
Answer:
B. Ask for a re-design of the hierarchy with changing
  the Operating System to a field rather than Class
  type

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

Question: 81
What will happen when you attempt to compile
and run this code?

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

        }
}
```

A. The code will compile and run, printing out the words
  "My Func"
B. The compiler will complain that the Base class has non
  abstract methods
C. The code will compile but complain at run time that
  the Base class has non abstract methods
D. The compiler will complain that the method myfunc
  in the base class has no body, nobody at all to looove it
  Question Help.
Answer:
A. The code will compile and run, printing out the
  words "My Func"

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

Question: 82
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: 83
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: 84
What will happen when you attempt to compile and run the following code:-

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

A. Compilation succeeds and at run time
   an output of 0 and false
B. Compilation succeeds and at run time
   an output of 0 and true
C. Compile time error b is not initialised
D. Compile time error z must be assigned
   a char value
   Question Help.

Answer:
A. Compilation succeeds and at run time an output
   of 0 and false.

Explanation:
The default value for a boolean declared at class level is false, and integer is 0;

Question: 85
Which of the following are collection classes

1) Collection
2) Iterator
3) HashSet
4) Vector
Question Help.
3) HashSet
4) Vector
The others are interfaces not classes

Question: 86
Which of the following most closely describes a bitset collection?

A. A class that contains groups of unique sequences of bits
B. A method for flipping individual bits in instance of a primitive type
C. An array of boolean primitives that indicate zeros or ones
D. A collection for storing bits as on-off information, like a vector of bits
Question Help.
Answer:
D. A collection for storing bits as on-off information, like a vector of bits.

Explanation:
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.

Question: 87
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.

A. Set the gridy value of the GridBagConstraint class to a value increasing from 1 to 4
B. Set the fill value of the GridBagConstrint class to VERTICAL
C. Set the ipady value of the GridBagConstraint class to a value increasing from 0 to 4
D. Set the fill value of the GridBagLayout class to GridBag.VERTICAL
Question Help.
Answer:
A. Set the gridy value of the GridBagConstraint class to a value increasing from 1 to 4

Explanation:
Answer D. 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.

Question: 88
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: 89
Which of the following statements are true?

1) All of the variables in an interface are implicitly static
2) All of the variables in an interface are implicitly final
3) All of the methods in an interface are implictly abstract
4) Any methods in an interface cannot access class level variables
Question Help.
1) All of the variables in an interface are implicitly static
2) All of the variables in an interface are implicitly final
3) All of the methods in an interface are implictly abstract

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

Question: 90
Which of the following statements are true?

1) Constructors cannot have a visibility modifier

2) Constructors can be marked public and protected,
  but not private

3) Constructors can only have a primitive return type

4) Constructors are not inherited
   Question Help.
4) Constructors are not inherited

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

Question: 91
What will be output by the following line of code?

```
System.out.println(010|4);
```

A. 14
B. 0
C. 6
D. 12
   Question Help.
Answer:
D. 12

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

8|4

To convert this to binary means

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

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

Question: 92
Which are keywords in Java?
A. NULL
B. sizeof
C. friend
D. extends
E. synchronized
Question Help.
Answer:
D. extends
E. synchronized

Question: 93
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: 94
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
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: 95
Which of the following statements are true?

1) Constructors are not inherited
2) Constructors can be overriden
3) A parental constructor can be invoked using this
4) Any method may contain a call to this or super
Question Help.
1) Constructors are not inherited

Question: 96
Which of the following statements are correct?

A. If multiple listeners are added to a component
   only events for the last listener added will be
   processed
B. If multiple listeners are added to a component
   the events will be processed for all but with no
   guarantee in the order
C. Adding multiple listeners to a comnponent will
   cause a compile time error
D. You may remove as well add listeners to a
   component.
   Question Help.
Answer:
B. If multiple listeners are added to a component
   the events will be processed for all but with no
   guarantee in the order
D. You may remove as well add listeners to a
   component.

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

Question: 97
How do you change the current layout manager
for a container

A. Use the setLayout method
B. Once created you cannot change the current layout manager of
   a component
C. Use the setLayoutManager method
D. Use the updateLayout method
   Question Help.
Answer:
A. Use the setLayout method

Question: 98
Given the following code

```
class Base{}

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

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

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

A. b=m;
B. m=b;
C. d =i;
D. b1 =i;
   Question Help.
Answer:
A. b=m;
C. d =i;

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

Question: 99
How does the set collection deal with duplicate elements?

A. An exception is thrown if you attempt to add
   an element with a duplicate value
B. The add method returns false if you attempt
   to add an element with a duplicate value
C. A set may contain elements that return duplicate
   values from a call to the equals method
D. Duplicate values will cause an error at compile time
   Question Help.
Answer:
B. The add method returns false if you attempt to
   add an element with a duplicate value

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

Question: 100
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?

A. Set
B. List
C. Map
D. Vector
   Question Help.

Answer:
A. 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.

Question: 101
Which of the following will compile without error?

A. File f = new File("/","autoexec.bat");
B. DataInputStream d = new DataInputStream(System.in);
C. OutputStreamWriter o = new OutputStreamWriter(System.out);
D. RandomAccessFile r = new RandomAccessFile("OutFile");
Question Help.
Answer:
A. File f = new File("/","autoexec.bat");
B. DataInputStream d = new DataInputStream(System.in);
C. OutputStreamWriter o = new OutputStreamWriter(System.out);

Explantion:
D., with the RandomAccess file will not compile
because the constructor must also be passed a mode
parameter which must be either "r" or "rw"

Note:
Although the objectives do not specifically mention the need
to understand the I/O Classes, feedback from people who have
taken the exam indicate that you will get questions on this
topic. As you will probably need to know this in the real world
of Java programming, get familiar with the basics. I have
assigned these questions to Objective 10 as that is a fairly
vague objective.

Question: 102
For a class defined inside a method, what rule governs
access to the variables of the enclosing method?

A. The class can access any variable
B. The class can only access static variables
C. The class can only access transient variables
D. The class can only access final variables
Question Help.
Answer:
D. The class can only access final variables

Question: 103
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?

A. 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.

B. 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.

C. 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.

D. 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.
Question Help.

Answer:

B. 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.

Explanation:
The Key to using the GridBagLayout manager is the
GridBagConstraint 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).

Question: 104
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?

A. 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.

B. 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.

C. 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.

D. 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.
Question Help.

Answer:
B. 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.

Explanation:
The Key to using the GridBagLayout manager is the
GridBagConstraint 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).

Question: 105
Given the following main method in a class
called Cycle and a command line of:-

```
java Cycle one two
```

what will be output?

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

A. None of these options
B. cycle
C. one
D. two
Question Help.

Answer:
C. one

Explanation:
Command line parameters start from 0 and from
the first parameter after the name of the compile
(normally Java).

Question: 106

Which of the following are Java keywords?

1) sizeof
2) main
3) transient
4) volatile
   Question Help.
3) transient
4) volatile

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

Question: 107
What can contain objects that have a unique key field of String type, if it is required to retrieve the objects using that key field as an index?

A. Map
B. Set
C. List
D. Collection
E. Enumeration
   Question Help.
Answer:
A. Map

Question: 108
Which most closely matches a description of a Java Map?

A. A vector of arrays for a 2D geographic representation
B. A class for containing unique array elements
C. A class for containing unique vector elements
D. An interface that ensures that implementing classes
   cannot contain duplicates
   Question Help.
Answer:
D. An interface that ensures that implementing classes
   cannot contain duplicates

Question: 109
You want to find out the value of the last element of an array. You write the following code. What will

happen when you compile and run it.?

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

A. An error at compile time
B. An error at run time
C. The value 0 will be output
D. The string "null" will be output
  Question Help.
Answer:
B. An error at run time

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

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

Question: 110
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: 111
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: 112
Which of the following statements are true

A. An inner class may be defined as static
B. An inner class may NOT be define as private
C. An anonymous class may have only one constructor
D. An inner class may extend another class
  Question Help.
Answer:
A. An inner class may be defined as static
D. An inner class may extend another class

Explanation:
How could an anonymous class have a constructor?.
Remember a constructor is a method with no return
type and the same name as the class. Inner classes
may be defined as private

Question: 113
Which of the following are Java modifiers?

A.public
B. private
C. friendly
D. transient
E. vagrant
  Question Help.
Answer:
A. public
B. private
D. transient

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

Question: 114
Which of the following are true?

1) A component may have only one event listener attached at a time

2) An event listener may be removed from a component
3) The ActionListener interface has no corresponding Adapter class
4) The processing of an event listener requires a try/catch block

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

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

Question: 115
Which of the following will compile without error?

1)

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

2)

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

3)

```
int i =1;
System.out.println(i1);
```

4)

```
int i =1;
System.out.println(i1);
```

Question Help.
1)

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

4)

```
int i =1;
System.out.println(i1);
```

Be aware that Integer (not the upper case I) is a wrapper class and thus cannot be treated like a primitive.
The fact that option 1 will compile may be a surprise, but although the char type is normally used to store character types, it is

actually an unsigned integer type. The reason option 3 does not compile
is that Java has a >>> operator but not a operator.

Question: 116
Which of the following statements are true?

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

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

Question: 117
Which of the following statements are true?

1) An interface can only contain methods and not variables

2) Java does not allow the creation of a reference to an
  interface with the new keyword.

3) A class may extend only one other class and implement
 only one interface

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

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

Question: 118
Which of the following statements are true?

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

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

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

Question: 119
Which statement is true about a non-static inner class?

A. It must implement an interface.
B. It is accessible from any other class.
C. It can only be instantiated in the enclosing class.
D. It must be final if it is declared in a method scope.
E. It can access private instance variables in the enclosing object.
Question Help.
Answer:
E. It can access private instance variables in the enclosing object.

Question: 120
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: 121
What will happen when you attempt to compile and run this code?

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

A. Error at compilation: native method cannot be static
B. Error at compilation native method must return value
C. Compilation but error at run time unless you have made code containing native amethod available
D. Compilation and execution without error
Question Help.
Answer:
D. Compilation and execution without error

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

Question: 121
What will happen when you attempt to compile and run this code?

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

A. Error at compilation: native method cannot be static
B. Error at compilation native method must return value
C. Compilation but error at run time unless you have made code containing native amethod available
D. Compilation and execution without error
Question Help.
Answer:
D. Compilation and execution without error

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

Question: 123
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: 124
Which of the following are Java keywords?

1) NULL
2) new
3) instanceOf
4) wend
   Question Help.
2) new

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

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

Question: 125
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: 126
What will happen when you attempt to compile and run this code?

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

A. Compile time errror indicating that a class with any
   final methods must be declared final itself
B. Compile time error indicating that you cannot inherit
   from a class with final methods
C. Run time error indicating that Base is not defined as
   final
D. Success in compilation and output of "amethod" at run
   time.
   Question Help.
Answer:
D. Success in compilation and output of "amethod" at
   run time.

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

Question: 127
Which of the following statements are true?

1) static methods do not have access to the implicit variable called this
2) a static method may not be overriden
3) a static method may not be overriden to be non-static
4) a static method may not be overloaded
   Question Help.
1) static methods do not have access to the implicit variable called this
3) a static may not be overriden to be non-static

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

Question: 128
What will happen when you attempt to compile and run the following class?

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

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

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

Question Help.
2) Compile time error

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

Question: 129
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: 130
You wish to store a small amount of data and make it available for rapid access. You do not have a need for the data to be sorted, uniqueness is not an issue and the data will remain fairly static. Which data structure might be most suitable for this requirement?

1) TreeSet
2) HashMap
3) LinkedList

4) an array
   Question Help.
4) an array

If you are only need to simply store a small amount of static data then you may as well go for the simple solution of an array. You might consider that you could future proof yourself against changing requirements by using one of the classes that offer wider functionality, but that is outside the scope of this question.