1) You want subclasses in any package to have access to members of a superclass. Which is the most restrictive access that accomplishes this objective?

A. public B. private **C. protected** D. transient

2) public class Outer

{

public void someOuterMethod()

{

//Line 5

}

public class Inner { }

public static void main(String[] argv)

{

Outer ot = new Outer();

//Line 10

}

}

Which of the following code fragments inserted, will allow to compile?

**A. new Inner();** //At line 5 B. new Inner(); //At line 10

C. new ot.Inner(); //At line 10 D. new Outer.Inner(); //At line 10

3) interface Base

{

boolean m1 ();

byte m2(short s);

}

Which two code fragments will compile?

1. interface Base2 implements Base {}
2. abstract class Class2 extends Base

{ public boolean m1(){ return true; }}

1. abstract class Class2 implements Base {}
2. abstract class Class2 implements Base

{ public boolean m1(){ return (7 > 4); }}

1. abstract class Class2 implements Base

{ protected boolean m1(){ return (5 > 7) }}

A. 1 and 2 B. 2 and 3 **C. 3 and 4** D. 1 and 5

5) public class Test { }

What is the prototype of the default constructor?

1. Test( ) B. Test(void) **C. public Test( )** D. public Test(void)

6) What is the most restrictive access modifier that will allow members of one class to have access to members of another class in the same package?

A. public B. abstract C. protected D. synchronized E. default access

7) Which of the following is/are legal method declarations?

1. protected abstract void m1();

2. static final void m1(){}

3. synchronized public final void m1() {}

4. private native void m1();

A. 1 and 3 B. 2 and 4 C. 1 only **D. All of them are legal declarations.**

9) Which three are valid method signatures in an interface?

1. private int getArea();

2. public float getVol(float x);

3. public void main(String [] args);

4. public static void main(String [] args);

5. boolean setFlag(Boolean [] test);

A. 1 and 2 **B. 2, 3 and 5** C.3, 4, and 5 D. 2 and 4

10) You want a class to have access to members of another class in the same package. Which is the most restrictive access that accomplishes this objective?

A. public B. private C. protected **D. default access**

11) What is the widest valid returnType for methodA in line 3?

public class ReturnIt

{

returnType methodA(byte x, double y) /\* Line 3 \*/

{

return (long)x / y \* 2;

}

}

A.int B. byte C. long **D. double**

12) class A

{

protected int method1(int a, int b)

{

return 0;

}

}

Which is valid in a class that extends class A?

A. **public int method1(int a, int b) {return 0; }**

B. private int method1(int a, int b) { return 0; }

C. public short method1(int a, int b) { return 0; }

D. static protected int method1(int a, int b) { return 0; }

14) Which two of the following are legal declarations for non-nested classes and interfaces?

1. final abstract class Test {} 2. public static interface Test {}

3. final public class Test {} 4. protected abstract class Test {}

5.protected interface Test {} 6. abstract public class Test {}

A. 1 and 4 B. 2 and 5 **C. 3 and 6** D. 4 and 6

15) Which of the following class level (nonlocal) variable declarations will not compile?

A. protected int a; B. transient int b = 3; **C. private synchronized int e;**

D. volatile int d;

16) Given a method in a protected class, what access modifier do you use to restrict access to that method to only the other members of the same class?

A. final B. static **C. private** D. protected E. volatile

17) Which is a valid declaration within an interface?

**A. public static short stop = 23;**

B. protected short stop = 23;

C. transient short stop = 23;

D. final void madness(short stop);

18) What will be the output of the program?

class A

{

final public int GetResult(int a, int b) { return 0; }

}

class B extends A

{

public int GetResult(int a, int b) {return 1; }

}

public class Test

{

public static void main(String args[])

{

B b = new B();

System.out.println("x = " + b.GetResult(0, 1));

}

}

A. x = 0 B. x = 1 **C. Compilation fails.** D. An exception is thrown at runtime.

19) What will be the output of the program?

public class Test

{

public static void main(String args[])

{

class Foo

{

public int i = 3;

}

Object o = (Object)new Foo();

Foo foo = (Foo)o;

System.out.println("i = " + foo.i);

}

}

1. **i = 3** B. Compilation fails. C. i = 5 D. ClassCastException will occur.

20) What will be the output of the program?

public class A

{

void A() /\* Line 3 \*/

{

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

}

public static void main(String[] args)

{

new A();

}

}

A.Class A B.Compilation fails. C. An exception is thrown at line 3.

D. **The code executes with no output.**

21) What will be the output of the program?

class Super

{

public int i = 0;

public Super(String text) /\* Line 4 \*/

{

i = 1;

}

}

class Sub extends Super

{

public Sub(String text)

{

i = 2;

}

public static void main(String args[])

{

Sub sub = new Sub("Hello");

System.out.println(sub.i);

}

}

A. 0 B. 1 C. 2 **D. Compilation fails.**

22) What will be the output of the program?

public class Test

{

public int aMethod()

{

static int i = 0;

i++;

return i;

}

public static void main(String args[])

{

Test test = new Test();

test.aMethod();

int j = test.aMethod();

System.out.println(j);

}

}

A. 0 B. 1 C. 2 **D. Compilation fails.**

23) What will be the output of the program?

interface Count

{

short counter = 0;

void countUp();

}

public class TestCount implements Count

{

public static void main(String [] args)

{

TestCount t = new TestCount();

t.countUp();

}

public void countUp()

{

for (int x = 6; x>counter; x--, ++counter) /\* Line 14 \*/

{

System.out.print(" " + counter);

}

}

}

A. 0 1 2 B. 1 2 3 C. 0 1 2 3 D. 1 2 3 4 **E. Compilation fails**

24) What will be the output of the program?

class Base

{

Base()

{

System.out.print("Base");

}

}

public class Alpha extends Base

{

public static void main(String[] args)

{

new Alpha(); /\* Line 12 \*/

new Base(); /\* Line 13 \*/

}

}

A. Base **B. BaseBase** C. Compilation fails D. The code runs with no output

25) What will be the output of the program?

public class NewTreeSet2 extends NewTreeSet

{

public static void main(String [] args)

{

NewTreeSet2 t = new NewTreeSet2();

t.count();

}

}

protected class NewTreeSet //Line 10

{

void count()

{

for (int x = 0; x < 7; x++,x++ )

{

System.out.print(" " + x);

}

}

}

A. 0 2 4 B. 0 2 4 6 C. Compilation fails at line 2 **D. Compilation fails at line 10**

26) What will be the output of the program?

public class ArrayTest

{

public static void main(String[ ] args)

{

float f1[ ], f2[ ];

f1 = new float[10];

f2 = f1;

System.out.println("f2[0] = " + f2[0]);

}

}

**A. It prints f2[0] = 0.0** B. It prints f2[0] = NaN

C. An error at f2 = f1; causes compile to fail.

D. It prints the garbage value.

27) What will be the output of the program?

class Super

{

public Integer getLength()

{

return new Integer(4);

}

}

public class Sub extends Super

{

public Long getLength()

{

return new Long(5);

}

public static void main(String[] args)

{

Super sooper = new Super();

Sub sub = new Sub();

System.out.println(

sooper.getLength().toString() + "," + sub.getLength().toString() );

}

}

A. 4, 4 B. 4, 5 C. 5, 4 **D. Compilation fails.**

28) interface DoMath

{

double getArea(int rad);

}

interface MathPlus

{

double getVol(int b, int h);

}

/\* Missing Statements ? \*/

Which two code fragments inserted at end of the program, will allow to compile?

1. class AllMath extends DoMath { double getArea(int r); }

2.interface AllMath implements MathPlus { double getVol(int x, int y); }

3.interface AllMath extends DoMath { float getAvg(int h, int l); }

4.class AllMath implements MathPlus { double getArea(int rad); }

5.abstract class AllMath implements DoMath, MathPlus

{ public double getArea(int rad) { return rad \* rad \* 3.14; } }

A. 1 only B. 2 only

**C. 3 and 5** D. 1 and 4

29) Which three statements are true?

1. The default constructor initializes method variables.

2. The default constructor has the same access as its class.

3. The default constructor invokes thedefault constructor of the superclass.

4. If a class lacks a no-arg constructor, the compiler always creates a default constructor.

5. The compiler creates a default constructor only when there are no other constructors for the class.

A. 1, 2 and 4 **B. 2, 3 and 5** C. 3, 4 and 5 D. 1, 2 and 3

30) package testpkg.p1;

public class ParentUtil

{

public int x = 420;

protected int doStuff() { return x; }

}

package testpkg.p2;

import testpkg.p1.ParentUtil;

public class ChildUtil extends ParentUtil

{

public static void main(String [] args)

{

new ChildUtil().callStuff();

}

void callStuff()

{

System.out.print("this " + this.doStuff() ); /\* Line 18 \*/

ParentUtil p = new ParentUtil();

System.out.print(" parent " + p.doStuff() ); /\* Line 20 \*/

}

}

which statement is true?

A. The code compiles and runs, with output this 420 parent 420.

B. If line 18 is removed, the code will compile and run.

C. **If line 20 is removed, the code will compile and run**.

D. An exception is thrown at runtime.

31) What is the expected output?

public class Profile {

private Profile(int w) {

System.out.println(w);

}

public static Profile() {

System.out.println(10);

}

public static void main(String args[]) {

Profile obj = new Profile(50);

}

}

A. 50 B. 10 C.Runtime exception **D. Compilation error stating illegal modifier for constructor.**

32) What is the result of compiling and running the following code?

class InitDemo{

static int i=demo();

static{System.out.println(i);}

InitDemo(){

System.out.println(“hello 1”);

}

public static void main(String... args){

System.out.println(“Hello 2”);

}

static int demo(){

System.out.println(“Inside Demo”);

return 10;

}

}

**A. Inside Demo**

**10**

**Hello 2**

B. Hello 2

Inside Demo

10

C.10

Hello 1

Hello 2

D. Compilation error

33) What is the result of compiling and running the following code?

public class Test {

static int p = test();

static public int test() {

System.out.println(p);

return 99;

}

public static void main(String[] args)

{

System.out.println(p);

}

}

**A.0 99** B. 99 0 C. 0 0 D. 99 99

34) What is the output of the following code?

public class Tester {

static int x = 4;

public Tester() {

System.out.print(this.x);

Tester();

}

public static void Tester() {

System.out.print(this.x);

}

public static void main(String... args) {

new Tester();

}

}

A. 4 4 B. 0 4 C. 4 0 **D. Compilation error stating cannot use this in static context.**

35) You have two packages, trunk1 and trunk2 where class Sheet declared in trunk1 and class Container declared in trunk2. Will the code compile? If not make it compile by doing the necessary modification. (Package should not be changed)

package trunk1;

public class Sheet {

public static int pageNumber = 99;

Sheet() {}

}

package trunk2;

import trunk1.Sheet;

public class Container {

public static void main(String... args) {

System.out.print(Sheet.pageNumber);

Sheet sheet = new Sheet();

}

}

1. The default constructor of class Sheet should be removed.
2. Default constructor should be defined in class Container.
3. **Change the access modifier of default constructor in class Sheet as public**.
4. It is not possible to make the code compile.