Q. What will be the result of attempting to compile this code?

```
import java.util.*;

package com.acme.toolkit;

public class AClass {
  public Other anInstance;
}

class Other {
  int value;
}
```

Select the one correct answer.
(a) The code will fail to compile, since the class Other has not yet been declared when referenced in the class AClass .
(b) The code will fail to compile, since an import statement cannot occur as the first statement in a source file.
(c) The code will fail to compile, since the package declaration cannot occur after an import statement.
(d) The code will fail to compile, since the class Other must be defined in a file called Other.java .
(e) The code will fail to compile, since the class Other must be declared public .
(f) The class will compile without errors.

Q.

Given the following code:

```
// (1) INSERT ONE IMPORT STATEMENT HERE
public class RQ700_20 {
  public static void main(String[] args) {
    System.out.println(sqrt(49));
  }
}
```

Which statements, when inserted at (1), will result in a program that prints 7, when compiled and run?

Select the two correct answers.

(a) `import static Math.*;`
(b) `import static Math.sqrt;`
(c) `import static java.lang.Math.sqrt;`
(d) `import static java.lang.Math.sqrt();`
(e) `import static java.lang.Math.*;`

Q.

Given the following code:

```
package p1;
enum Signal {
  GET_SET, ON_YOUR_MARKS, GO;
}
-------------------------------------------------------------
package p2;
// (1) INSERT IMPORT STATEMENT(S) HERE
public class RQ700_50 {
  public static void main(String[] args) {
    for(Signal sign : Signal.values()) {
      System.out.println(sign);
    }
  }
}
```

Which import statement(s), when inserted at (1), will result in a program that prints the constants of the enum type `Signal`, when compiled and run?

Select the one correct answer.

(a) `import static p1.Signal.*;`

(b) `import p1.Signal;`

(c) `import p1.*;`

(d) `import p1.Signal;`
`import static p1.Signal.*;`

(e) `import p1.*;`
`import static p1.*;`

(f) None of the above.

Q. Given the following code:

```
package p3;
public class Util {
  public enum Format {
    JPEG { public String toString() {return "Jpeggy"; }},
    GIF  { public String toString() {return "Giffy"; }},
    TIFF { public String toString() {return "Tiffy"; }};
  }
  public static <T> void print(T t) {
    System.out.print("|" + t + "|");
  }
}
-------------------------------------------------------------
// (1) INSERT IMPORT STATEMENTS HERE
public class NestedImportsA {
  public static void main(String[] args) {
    Util u = new Util();
    Format[] formats = {
        GIF, TIFF,
        JPEG,
        Format.JPEG,
        Util.Format.JPEG,
        p3.Util.Format.JPEG
```

```
    };
    for (Format fmt : formats)
      print(fmt);
  }
}
```

Which sequence of import statements, when inserted at (1), will result in the code compiling, and the execution of the `main()` method printing:

```
|Giffy||Tiffy||Jpeggy||Jpeggy||Jpeggy||Jpeggy|
```

Select the three correct answers.

(a)
```
import p3.Util;
import p3.Util.Format;
import static p3.Util.print;
import static p3.Util.Format.*;
```

(b)
```
import p3.Util;
import static p3.Util.Format;
import static p3.Util.print;
import static p3.Util.Format.*;
```

(c)
```
import p3.*;
import static p3.Util.*;
import static p3.Util.Format.*;
```

(d)
```
import p3.*;
import p3.Util.*;
import static p3.Util.Format.*;
```

Q.

Which statements are true about the import statement?

Select the two correct answers.

(a) Static import from a class automatically imports names of static members of any nested types declared in that class.
(b) Static members of the default package cannot be imported.
(c) Static import statements must be specified after any type import statements.
(d) In the case of a name conflict, the name in the last static import statement is chosen.
(e) A declaration of a name in a compilation unit can shadow a name that is imported.

Q. Given the following class, which of these alternatives are valid ways of referring to the class from outside of the package net.basemaster ?

```
package net.basemaster;
public class Base {
     // ...
}
```

Select the two correct answers.

A. By simply referring to the class as Base .
B. By simply referring to the class as basemaster.Base .
C. By simply referring to the class as net.basemaster.Base .
D. By importing with net.basemaster.* , and referring to the class as Base .
E. By importing with net.* , and referring to the class as basemaster.Base .

Q.Which one of the following class declarations is a valid declaration of a class that cannot be instantiated? Select the one correct answer.
(a) class Ghost { abstract void haunt(); }
(b) abstract class Ghost { void haunt(); }
(c) abstract class Ghost { void haunt() {}; }
(d) abstract Ghost { abstract void haunt(); }
(e) static class Ghost { abstract haunt(); }

Q. Which one of the following class declarations is a valid declaration of a class that cannot be extended? Select the one correct answer.
(a) class Link { }
(b) abstract class Link { }
(c) native class Link { }
(d) static class Link { }
(e) final class Link { }
(f) private class Link { }
(g) abstract final class Link { }

Q.Given the following declaration of a class, which fields are accessible from outside the package com.corporation.project ?

```
package com.corporation.project;
public class MyClass {
     int i;
     public
     int j;
     protected int k;
```

```
        private
        int l;
    }
```

Select the two correct answers.
a)Field i is accessible in all classes in other packages.
b)Field j is accessible in all classes in other packages.
c)Field k is accessible in all classes in other packages.
d)Field k is accessible in subclasses only in other packages.
e)Field l is accessible in all classes in other packages.
f)Field l is accessible in subclasses only in other packages.

Q. How restrictive is the default accessibility compared to public , protected , and private accessibility?
Select the one correct answer.

a)Less restrictive than public .
b)More restrictive than public , but less restrictive than protected .
c)More restrictive than protected , but less restrictive than private .
d)More restrictive than private .
e)Less restrictive than protected from within a package, and more restrictive than protected from outside a package.

Q. Which statement is true about the accessibility of members?
Select the one correct answer.
(a) A private member is always accessible within the same package.
(b) A private member can only be accessed within the class of the member.
(c) A member with default accessibility can be accessed by any subclass of the class in which it is declared.
(d) A private member cannot be accessed at all.
(e) Package/default accessibility for a member can be declared using the key-word default .

Q. Which lines that are marked will compile in the following code?

```
//Filename: A.java
package packageA;

public class A {
  protected int pf;
}
```

```
//Filename: B.java
package packageB;
import packageA.A;

public class B extends A {
  void action(A obj1, B obj2, C obj3) {
    pf = 10;                  // (1)
    obj1.pf = 10;             // (2)
    obj2.pf = 10;             // (3)
    obj3.pf = 10;             // (4)

  }
}

class C extends B {
  void action(A obj1, B obj2, C obj3) {
    pf = 10;                  // (5)
    obj1.pf = 10;             // (6)
    obj2.pf = 10;             // (7)
    obj3.pf = 10;             // (8)
  }
}

class D {
  void action(A obj1, B obj2, C obj3) {
    pf = 10;                  // (9)
    obj1.pf = 10;             // (10)
    obj2.pf = 10;             // (11)
    obj3.pf = 10;             // (12)
  }
}
```

Select the five correct answers.

(a) (1)

(b) (2)

(c) (3)

(d) (4)
(e) (5)
(f) (6)
(g) (7)
(h) (8)
(i) (9)
(j) (10)
(k) (11)
(l) (12)

Q. Which statements about the use of modifiers are true?
Select the two correct answers.
(a) If no accessibility modifier (public , protected , or private) is specified for a member declaration, the member is only accessible by classes in the package of its class and by subclasses of its class in any package.
(b) You cannot specify accessibility of local variables. They are only accessible within the block in which they are declared.
(c) Subclasses of a class must reside in the same package as the class they extend.
(d) Local variables can be declared static .
(e) The objects themselves do not have any accessibility modifiers, only the object references do.

Q. Given the following source code, which comment line can be uncommented without introducing errors?

```
abstract class MyClass {
  abstract void f();
  final    void g() {}
//final    void h() {}                          // (1)

  protected static int i;
  private          int j;
}

final class MyOtherClass extends MyClass {
//MyOtherClass(int n) { m = n; }                // (2)

  public static void main(String[] args) {
    MyClass mc = new MyOtherClass();
  }

  void f() {}
  void h() {}
//void k() { i++; }                             // (3)
//void l() { j++; }                             // (4)

  int m;
}
```

Select the one correct answer.

(a) (1)
(b) (2)
(c) (3)
(d) (4)

Q. What would be the result of compiling and running the following program?

```
class MyClass {
  static MyClass ref;
  String[] arguments;

  public static void main(String[] args) {
    ref = new MyClass();
    ref.func(args);
  }

  public void func(String[] args) {
    ref.arguments = args;
  }
}
```

Select the one correct answer.

(a) The program will fail to compile, since the static method main() cannot have a call to the non-static method func() .
(b) The program will fail to compile, since the non-static method func() cannot access the static variable ref .
(c) The program will fail to compile, since the argument args passed to the static method main() cannot be passed to the non-static method func() .
(d) The program will compile, but will throw an exception when run.
(e) The program will compile and run successfully

Q. Given the following member declarations, which statement is true?

```
int a;                          // (1)
static int a;                   // (2)
int f() { return a; }           // (3)
static int f() { return a; }    // (4)
```

Select the one correct answer.

a)Declarations (1) and (3) cannot occur in the same class declaration.
b)Declarations (2) and (4) cannot occur in the same class declaration.
c)Declarations (1) and (4) cannot occur in the same class declaration.
d)Declarations (2) and (3) cannot occur in the same class declaration.

Q. Which statement is true?
Select the one correct answer.
(a) A static method can call other non-static methods in the same class by using the this keyword.
(b) A class may contain both static and non-static variables, and both static and non-static methods.
(c) Each object of a class has its own instance of the static variables declared in the class.
(d) Instance methods may access local variables of static methods.
(e) All methods in a class are implicitly passed the this reference as argument, when invoked.
Q. What, if anything, is wrong with the following code?

```
abstract class MyClass {
     transient int j;
     synchronized int k;
     final void MyClass() {}
     static void f() {} }
```

Select the one correct answer.
a)The class MyClass cannot be declared abstract .
b)The field j cannot be declared transient .
c)The field k cannot be declared synchronized .
d)The method MyClass() cannot be declared final .
e)The method f() cannot be declared static .
f)Nothing is wrong with the code; it will compile successfully.

Q.Which one of these is not a legal member declaration within a class?
Select the one correct answer.
(a) static int a;
(b) final Object[] fudge = { null };
(c) abstract int t;
(d) native void sneeze();
(e) final static private double PI = 3.14159265358979323846;

Q. Which statements about modifiers are true?
Select the two correct answers.
a)Abstract classes can declare final methods.
b)Fields can be declared native .
c)Non- abstract methods can be declared in abstract classes.
d)Classes can be declared native .
e)Abstract classes can be declared final .

Q. Which statement is true?
Select the one correct answer.
(a) The values of transient fields will not be saved during serialization.
(b) Constructors can be declared abstract .
(c) The initial state of an array object constructed with the statement int[] a = new int[10] will depend on whether the array variable a is a local variable or a field.
(d) A subclass of a class with an abstract method must provide an implementation for the abstract method.
(e) Only static methods can access static members.