Exam 3: QUESTIONS

Question 1 of 5

Given the following,

```
1. interface Base {
2.   boolean m1 ();
3.   byte m2(short s);
4. }
```

Which code fragments will compile? (Choose all that apply.)

A.
```
interface Base2 implements Base { }
```

B.
```
abstract class Class2 extends Base {
  public boolean m1() { return true; } }
```

C.
```
abstract class Class2 implements Base { }
```

D.
```
abstract class Class2. implements Base {
  public boolean m1() { return (true); } }
```

E.
```
class Class2 implements Base {
  boolean m1( ) { return false; }
  byte m2(short s) { return 42; } }
```

Question 2 of 5

Which declare a compilable abstract class?

(Choose all that apply.)

```
A.public abstract class Canine { public Bark speak(); }
B.public abstract class Canine { public Bark speak() { } }
C.public class Canine { public abstract Bark speak(); }
D.public class Canine abstract { public abstract Bark speak(); }
```

Question 3 of 5

Which is true? (Choose all that apply.)

A. "X extends Y" is correct if and only if X is a class and Y is an interface.
B. "X extends Y" is correct if and only if X is an interface and Y is a class.
C. "X extends Y" is correct if X and Y are either both classes or both interfaces.
D. "X extends Y" is correct for all combinations of X and Y being classes and/or interfaces.

Question 4 of 5

Given:

```
1. class Voop {
2.   public static void main(String [] args) {
3.     doStuff(1);
4.     doStuff(1, 2);
5.   }
6.   // insert code here
7. }
```

Which, inserted independently at line 6, will compile? (Choose all that apply.)

A. static void doStuff(int... doArgs) { }
B. static void doStuff (int [] doArgs) { }
C. static void doStuff(int doArgs...) { }
D. static void doStuff(int... doArgs, int y) { }
E. **static void doStuff(int x, int... doArgs) { }**

Question 5 of 5

Given:

```
1. enum Animals {
2.   DOG ("woof"), CAT ("meow"), FISH ("burble");
3.   String sound;
4.   Animals(String s) { sound = s; }
5. }
6. class TestEnum {
7.   static Animals a;
8.   public static void main(String[] args) {
9.     System.out.println(a.DOG.sound + " " + a.FISH.sound);
10.  }
11. }
```

What is the result?

A. woof burble
B. Multiple compilation errors
C. Compilation fails due to an error on line 2
D. Compilation fails due to an error on line 3
E. Compilation fails due to an error on line 4
F. Compilation fails due to an error on line 9

SOLUTIONS

Question 1 of 5

Given the following,

```
1. interface Base {
2.   boolean m1 ();
3.   byte m2(short s);
4. }
```

Which code fragments will compile? (Choose 2 options.)

A.
```
interface Base2 implements Base { }
```

B.
```
abstract class Class2 extends Base {
  public boolean m1() { return true; } }
```

C.
```
abstract class Class2 implements Base { }
```

D.
```
abstract class Class2 implements Base {
  public boolean m1() { return (true); } }
```

E.
```
class Class2 implements Base {
  boolean m1( ) { return false; }
  byte m2(short s) { return 42; } }
```

1. ☑ **C** and **D** are correct, **C** is correct because an abstract class doesn't have to implement any or all of its interface's methods. **D** is correct because the method is correctly implemented.
   ☒ **A** is incorrect because interfaces don't implement anything, **B** is incorrect because classes don't extend interfaces. **E** is incorrect because interface methods are implicitly public, so the methods being implemented must be public. (Objective 1.1)

Question 2 of 5

Which declare a compilable abstract class?

(Choose 1 option)

```
A. public abstract class Canine { public Bark speak(); }
B. public abstract class Canine { public Bark speak() { } }
C. public class Canine { public abstract Bark speak(); }
D. public class Canine abstract { public abstract Bark speak(); }
```

☑ **B** is correct. abstract classes don't have to have any abstract methods.
☒ **A** is incorrect because abstract methods must be marked as such, **C** is incorrect because you can't have an abstract method unless the class is abstract. **D** is incorrect because the keyword abstract must come before the class name.

Question 3 of 5

Which is true? (Choose 1 option)

A. "X extends Y" is correct if and only if X is a class and Y is an interface.
B. "X extends Y" is correct if and only if X is an interface and Y is a class.
C. "X extends Y" is correct if X and Y are either both classes or both interfaces.
D. "X extends Y" is correct for all combinations of X and Y being classes and/or interfaces.

☑ **C** is correct.
☒ **A** is incorrect because classes implement interfaces, they don't extend them. **B** is incorrect because interfaces only "inherit from" other interfaces. **D** is incorrect based on the preceding rules.

Question 4 of 5

Given:

```
1. class Voop {
2.   public static void main(String [] args) {
3.     doStuff(1);
4.     doStuff(1, 2);
5.   }
6.   // insert code here
7. }
```

Which, inserted independently at line 6, will compile? (Choose 2 options.)

A. static void doStuff(int... doArgs) { }
B. static void doStuff (int [] doArgs) { }
C. static void doStuff(int doArgs...) { }
D. static void doStuff(int... doArgs, int y) { }
E. **static void doStuff(int x, int... doArgs) { }**

☑ **A** and **E** use valid var-args syntax.
☒ **B** and **C** are invalid var-arg syntax, and **D** is invalid because the var-arg must be the last of a method's arguments.

Question 5 of 5

Given:

```
1. enum Animals {
2.   DOG ("woof"), CAT ("meow"), FISH ("burble");
3.   String sound;
4.   Animals(String s) { sound = s; }
5. }
6. class TestEnum {
7.   static Animals a;
8.   public static void main(String[] args) {
9.     System.out.println(a.DOG.sound + " " + a.FISH.sound);
10.  }
11. }
```

What is the result? (Choose 1 option)

A. woof burble
B. Multiple compilation errors
C. Compilation fails due to an error on line 2
D. Compilation fails due to an error on line 3
E. Compilation fails due to an error on line 4
F. Compilation fails due to an error on line 9

☑ **A** is correct; enums can have constructors and variables.
☒ **B, C, D, E**, and **F** are incorrect; these lines all use correct syntax.