Started on	Tuesday, 17 May 2022, 12:27 PM
State	Finished
Completed on	Tuesday, 17 May 2022, 12:46 PM
Time taken	19 mins

Which statements are true for both abstract classes and interfaces? (Choose all that apply)

Select one or more:

- [ ] **a. Both can be extended using the extend keyword.**
- [x] **b. Both can contain public static final variables.**
- [ ] **c. All methods within them are assumed to be abstract.**
- [ ] **d. Neither can be instantiated directly.**
- [ ] **e. Both can contain default methods.**
- [ ] **f. Both can contain static methods.**
- [ ] **g. Both inherit java.lang.Object.**

The concept of default methods exists only in interfaces

What is the result of the following statements?

```
1: public class Test {
2: public void print(byte x) {
3: System.out.print("byte");
4: }
5: public void print(int x) {
6: System.out.print("int");
7: }
8: public void print(float x) {
9: System.out.print("float");
10: }
11: public void print(Object x) {
12: System.out.print("Object");
13: }
14: public static void main(String[] args) {
15: Test t = new Test();
16: short s = 123;
17: t.print(s);
18: t.print(true);
19: t.print(6.789);
20: }
21: }
```

Select one or more:

- [ ] **a. byteObjectObject**
- [ ] **b. bytefloatObject**
- [ ] **c. intObjectfloat**
- [x] **d. intObjectObject**
- [ ] **e. byteObjectfloat**
- [ ] **f. intfloatObject**

Choose the correct statement about the following code:

```
1: interface HasExoskeleton {
2: abstract int getNumberOfSections();
3: }
4: abstract class Insect implements HasExoskeleton {
5: abstract int getNumberOfLegs();
6: }
7: public class Beetle extends Insect {
8: int getNumberOfLegs() { return 6; }
9: }
```

Select one or more:

- [ ] **a. The code will not compile because of line 4.**
- [ ] **b. The code will not compile because of line 7.**
- [ ] **c. The code will not compile because of line 2.**
- [ ] **d. It compiles and runs without issue.**
- [ ] **e. It compiles but throws an exception at runtime.**

Choose the correct statement about the following code:

```
1: public interface CanFly {
2: void fly();
3: }
4: interface HasWings {
5: public abstract Object getWindSpan();
6: }
7: abstract class Falcon implements CanFly, HasWings {
8: }
```

If abstract class takes on to implement an interface, the class that extends the abstract class should take care of the methods of the interface also

Select one or more:

- [ ] **a. The code will not compile because of line 4.**
- [ ] **b. The code will not compile because of lines 2 and 5.**
- [ ] **c. It compiles without issue.**
- [ ] **d. The code will not compile because of line 5.**

- [ ] **e. The code will not compile because the class Falcon doesn't implement the interface methods.**
- [ ] **f. The code will not compile because of line 2.**

Which of the following statements about polymorphism are true? (Choose all that apply)

Select one or more:

- [x] **a. If a method takes a superclass of three objects, then any of those classes may be passed as a parameter to the method.**
- [ ] **b. By defining a public instance method in the superclass, you guarantee that the specific method will be called in the parent class at runtime.**
- [x] **c. All cast exceptions can be detected at compile-time.**
- [ ] **d. A method that takes a parameter with type java.lang.Object will take any reference.**
- [x] **e. A reference to an object may be cast to a subclass of the object without an explicit cast.**

Suppose that you need to work with a collection of elements that need to be sorted in their

natural order, and each element has a unique string associated with its value. Which of the

following collections classes in the java.util package best suit your needs for this scenario?

Select one or more:

- [ ] **a. HashMap**
- [x] **b. TreeMap**
- [ ] **c. HashSet**
- [ ] **d. TreeSet**
- [ ] **e. ArrayList**
- [ ] **f. Vector**

Which of the following can replace line 2 to make this code compile? (Choose all that apply)

```
1: import java.util.*;
2: // INSERT CODE HERE
3: public class Imports {
4: public void method(ArrayList<String> list) {
5: sort(list);
6: }
7: }
```

Select one or more:

- [ ] **a. static import java.util.Collections.*;**
- [x] **b. import static java.util.Collections.sort(ArrayList<String>);**
- [ ] **c. static import java.util.Collections;**
- [ ] **d. import static java.util.Collections;**
- [x] **e. import static java.util.Collections.*;**
- [ ] **f. static import java.util.Collections.sort(ArrayList<String>);**

What is the result of the following statements?

```
3: ArrayDeque<String> greetings = new ArrayDeque<String>();
4: greetings.push("hello");
5: greetings.push("hi");
6: greetings.push("ola");
7: greetings.pop();
8: greetings.peek();
9: while (greetings.peek() != null)
10: System.out.print(greetings.pop());
```

Select one or more:

- [ ] **a. hellohi**
- [ ] **b. hellohiola**
- [ ] **c. The code does not compile.**
- [x] **d. hihello**
- [ ] **e. An exception is thrown.**
- [ ] **f. hi**
- [ ] **g. hello**

Choose the correct statement about the following code:

```
1: public interface Herbivore {
2: int amount = 10;
3: public static void eatGrass();
4: public int chew() {
5: return 13;
6: }
7: }
```

Select one or more:

- [ ] **a. The code will not compile because of line 2.**
- [ ] **b. The code will not compile because of lines 3 and 4.**
- [ ] **c. The code will not compile because of line 4.**
- [ ] **d. The code will not compile because of line 3.**

- [ ] **e. The code will not compile because of lines 2 and 3.**
- [ ] **f. It compiles and runs without issue.**

What is the result of the following statements?

```
3: List list = new ArrayList();
4: list.add("one");
5: list.add("two");
6: list.add(7);
7: for (String s: list)
8: System.out.print(s);
```

Select one or more:

- [ ] **a. Compiler error on line 6**
- [x] **b. Compiler error on line 7**
- [ ] **c. onetwo**
- [ ] **d. onetwo7**
- [ ] **e. onetwo followed by an exception**

Which are true of the following code? (Choose all that apply)

```
1: public class Rope {
2: public static void swing() {
3: System.out.print("swing ");
4: }
5: public void climb() {
6: System.out.println("climb ");
7: }
8: public static void play() {
9: swing();
10: climb();
11: }
12: public static void main(String[] args) {
13: Rope rope = new Rope();
14: rope.play();
15: Rope rope2 = null;
16: rope2.play();
17: }
18: }
```

Select one or more:

- [ ] **a. If the lines with compiler errors are removed, the output is climb climb.**

- [ ] **b. If the lines with compile errors are removed, the code throws a NullPointerException.**
- [ ] **c. The code compiles as is.**
- [ ] **d. If the lines with compiler errors are removed, the output is swing swing.**
- [ ] **e. There is exactly one compiler error in the code.**
- [ ] **f. There are exactly two compiler errors in the code.**

Suppose that you have a collection of products for sale in a database and you need to display those products. The products are not unique. Which of the following collections classes in the java.util package best suit your needs for this scenario?

Select one or more:

- [ ] **a. HashMap**
- [ ] **b. ArrayList**
- [ ] **c. HashSet**
- [ ] **d. LinkedList**
- [ ] **e. Arrays**

How many compiler errors are in the following code?

```
1: public class RopeSwing {
2: private static final String leftRope;
3: private static final String rightRope;
4: private static final String bench;
5: private static final String name = "name";
6: static {
7: leftRope = "left";
8: rightRope = "right";
9: }
10: static {
11: name = "name";
12: rightRope = "right";
13: }
14: public static void main(String[] args) {
15: bench = "bench";
16: }
17: }
```

Select one or more:

- [ ] **a. 5**
- [ ] **b. 1**

- [ ] **c. 2**
- [ ] **d. 0**
- [ ] **e. 3**
- [x] **f. 4**

Which of these statements compile? (Choose all that apply.)

Select one or more:

- [ ] **a. HashSet<? super ClassCastException> set = new HashSet<Exception>();**
- [x] **b. List<Object> objects = new ArrayList<? extends Object>();**
- [ ] **c. List<Object> values = new HashSet<Object>();**
- [ ] **d. HashSet<Number> hs = new HashSet<Integer>();**
- [ ] **e. Map<String, ? extends Number> hm = new HashMap<String, Integer>();**
- [x] **f. List<String> list = new Vector<String>();**

What is the output of the following code?

```
import rope.*;
import static rope.Rope.*;
public class RopeSwing {
private static Rope rope1 = new Rope();
private static Rope rope2 = new Rope();
{
System.out.println(rope1.length);
}
public static void main(String[] args) {
rope1.length = 2;
rope2.length = 8;
System.out.println(rope1.length);
}
}
package rope;
public class Rope {
public static int length = 0;
}
```

Select one or more:

- [ ] **a. 02**
- [ ] **b. 8**
- [ ] **c. An exception is thrown.**
- [ ] **d. 2**

- [ ] **e. 08**
- [ ] **f. The code does not compile.**