Java 111 Final Exam Review (from version 1)

Total Questions: 25

Most Correct Answers: **#17**

Least Correct Answers: **#24**

1. **Instance variables are variables declared inside a method or method parameter.**

0/8 (A) True

0/8 (B) True only in an interface

0/8 (C) True only in an abstract class

7/8 (D) False

2. **In the code example below, which is the object reference variable?**

```
public class StackRef {

    public void run() {
        build();
    }

    public void build() {
        Duck d = new Duck(24);
    }
```

0/8 (A) Duck()

0/8 (B) 24

7/8 (C) d

0/8 (D) new Duck()

0/8 (E) Duck

3. **In the code example below, where will the Duck object live?**

```
public class StackRef {

    public void run() {
        build();
    }

    public void build() {
        Duck d = new Duck(24);
    }
```

7/8 (A) Heap

0/8 (B) Stack

0/8 (C) Frame

4. **The currently executing method is the one on the bottom of the stack.**

1/8 (A) True

6/8 (B) False

5. **In the code example below, how long will "d" live on the stack?**

```
18 public class StackRef {
19
20     public void run() {
21         Duck d = new Duck();
22         build();
23     }
24
25     public void build() {
26
27     }
28 }
```

3/8 (A) Until run() pops off the stack

3/8 (B) Until build() is added to the top of the stack

0/8 (C) Until build() pops off the stack

1/8 (D) Until d is garbage collected

6. A constructor on an object's parent must run when you make a new object.

5/8 A True

2/8 B False

7. Type the code that should appear on line 7 to call the Duck's single arg constructor with a parameter of 14.

```
1
2  public class Duck extends Animal {
3
4      int size;
5
6      public Duck() {
7
8      }
9
10     public Duck(int newSize) {
11         size = newSize;
12     }
```

Anon anon0780d85e98e34870

✓ this(14);

Anon anon0c1db66075c9447f

✗ Duck d = new Duck(12);

Anon anon34f2ed21335e4ace

✗ size size

Anon anon632270ffb36e4eb1

✗ Duck(int 0);

Anon anon689ea2fa741a46a9

✗ this.size(14)

Anon anona3f3d2a2e9fc4e4a

✓ this(14);

Anon anonbad1f43d4fee458d

✗ this(int newSize);

8. A constructor can have a call to super() OR this(), but NEVER both.

7/8 A True

1/8 B False

9. A class must be put into a directory structure that matches the package hierarchy.

7/8 A True

0/8 B False

10. The keyword "static" lets a method run without any instance of the class.

7/8 A True

1/8 B False

11. The following code will compile:

```
public class Cat {
int morningMeowVolume = 5;
int kittyCount;

public static int retrieveKittyCount() {
return kittyCount;
}
}
```

3/8 A True

4/8 B False

12. An abstract class can only have abstract methods.

3/8 A True

5/8 B False

13. Marking a class with the "abstract" keyword prevents a developer from instantiating that class.

7/8 A True

1/8 B False

14. Which of the following are true?

1/8 A An interface must be created using the keyword "abstract".

7/8 B An interface defines only abstract methods (prior to Java 8).

8/8 C A class can implement multiple interfaces.

7/8 D All interface methods are implicitly public.

0/8 E None of the above.

15. Write an abstract method called calculatePremium that accepts two parameters: a double for rate and an int for term. The method has returns a double.

Anon anon0780d85e98e34870

```
public abstract double calculatePremium(double rate, int term)
{
//double answer = do fancy magic maths
return answer;
}
```

✗

Anon anon0c1db66075c9447f

✓ public abstract double calculatePremium(double rate, int term);

Anon anon34f2ed21335e4ace

```
public double calculatePremium(){

System.out.println(122);
returns calcultePremium;
}
```

✗

Anon anon5713f8dc71814b33

✓ public abstract double calculatePremium(double rate, int term);

Anon anon632270ffb36e4eb1

✗ abstract double calculatePremium(double rate, int term) {}

Anon anon689ea2fa741a46a9

✓ public abstract double calculatePremium(double rate, int term);

Anon anona3f3d2a2e9fc4e4a

✗ public abstract double calculatePremuim(double rate, int term);

Anon anonbad1f43d4fee458d

✗ abstract double calculatePremium (double rate, int term) {}

16. The following will compile:

```
public class Puppy extends Dog {
public Puppy() {
setSize(.5);
super();
}
}
```

1/8 A True

7/8 B False

17. Given the following: SuperStarCoder rockstar = new JavaRockStar(); what is the object reference type?

0/8 A JavaRockStar

0/8 B rockstar

8/8 C SuperStarCoder

0/8 D new

0/8 E None of the above

18. What is the proper way to create an interface called Payable?

1/8 A public abstract interface class Payable {}

0/8 B public abstract Payable {}

0/8 C public abstract interface class Payable extends Payable {}

7/8 D public interface Payable {}

0/8 E None of the above

19. Which of the following are true?

8/8 A You can write a new instance method in the subclass that has the same signature as the one in the superclass, thus overriding it.

7/8 B You can declare new methods in the subclass that are not in the superclass.

7/8 C You can declare new fields in the subclass that are not in the superclass.

4/8 D A subclass cannot extend multiple superclasses.

2/8 E A subclass inherits the private instance variables and private methods of its superclass.

20. A superclass called Fruit contains a method called display() that outputs a message to the terminal. Which code segment IN THE SUBCLASS Apple will successfully call that method?

7/8 A super.display();

0/8 B Fruit.display();

0/8 C display(Fruit);

1/8 D Apple.display;

0/8 E this.display();

21. **Which of the following are true:**

6/8 A The arguments and return types of an overriding method must look to the outside world exactly like the overridden method in the superclass.

7/8 B Overloaded methods have the same name, but different argument lists.

6/8 C Overloaded methods can have different return types as long as the arguments lists are different.

3/8 D When overriding a public method, you can declare the method as private.

0/8 E All of the above are true.

22. **An import statement saves you from having to type out the full name of classes.**

5/8 A True

3/8 B False

23. **What is the proper way to add a Kumquat object to an ArrayList of Kumquats called myKumquatList? Assume I have created a Kumquat object like this: Kumquat myKumquat = new Kumquat();**

0/8 A myKumquatList.kumquat = myKumquat;

1/8 B myKumquatList[0] = myKumquat;

5/8 C myKumquatList.add(myKumquat);

1/8 D myKumquatList[1].add(myKumquat)

24. Event[] events = new Event[2];
events[0] = new Marathon();
events[1] = new Meeting();

If Marathon, Meeting and Event do not have a superclass/subclass relationship, explain how this code will compile and run.

Anon anon0780d85e98e34870

✗ They are of different types and won't compile?

Anon anon0c1db66075c9447f

✗ it won't compile because there is no object reference variable on the 2 objects

Anon anon34f2ed21335e4ace

✗ have the main method

Anon anon5713f8dc71814b33

✗ IDK :(

Anon anon632270ffb36e4eb1

✗ huh

Anon anon689ea2fa741a46a9

✗ idk

Anon anona3f3d2a2e9fc4e4a

✗ each of the classes must implement the same interface

25. Why is importing other packages or classes into your class necessary?

5/8 (A) so that the compiler can find the .class files of all classes referenced by your class.

2/8 (B) so that the compiler can find the source files of all classes reference by your class.

0/8 (C) so that the compiler can put your classes in the proper directory.

1/8 (D) so that the compiler knows where to find your source code.