Java　　　　　　　　　　　　　　　　　　　　　　　　　　　　Name -
Strings Multiple Choice Worksheet #1　　　　　　　　　　　　　Period -

1. Consider the declaration...

```
String line = "Some more silly stuff on strings!";
// the words are separated by a single space
```

What string will `str` refer to after execution of the following code segment?

```
int x = line.indexOf("m");
String str = line.substring(10, 15) + line.substring(25, 25 + x);
```

a. `"sillyst"`　　　　d. `"silly str"`
b. `"sillystr"`　　　e. `"sillystrin"`
c. `"silly st"`

2. A program has a `String` variable `fullName` that stores a first name, followed by a space, followed by a last name. There are no spaces in either the first or last names. Here are some examples of `fullName` values: `"Anthony Coppola"`, `"Jimmy Carroll"`, and `"Tom DeWire"`. Consider the following code segment that extracts the last name from a `fullName` variable, and stores it in `lastName` with no surrounding blanks:

```
int k = fullName.indexOf(" ")     // find index of blank
String lastName = /* expression */
```

Which is a correct replacement for `/* expression */`?

```
I.   fullName.substring(k);
II.  fullName.substring(k + 1);
III. fullName.substring(k + 1, fullName.length());
```

a.　`I` only　　　　　　d.　`II` and `III` only
b.　`II` only　　　　　e.　`I` and `III` only
c.　`III` only

3. One of the rules for converting English to Pig Latin states: If a word begins with a consonant, move the consonant to the end of the word and add "ay". Thus "dog" becomes "ogday", and "crisp" becomes "rispcay". Suppose `s` is a `String` containing an English word that begins with a consonant. Which of the following creates the correct corresponding word in Pig Latin? Assume the declarations

```
String ayString = "ay";
String pigString;
```

a. `pigString = s.substring(0, s.length()) + s.substring(0, 1) + ayString;`
b. `pigString = s.substring(1, s.length()) + s.substring(0, 0) + ayString;`
c. `pigString = s.substring(0, s.length() -1) + s.substring(0, 1) + ayString;`
d. `pigString = s.substring(1, s.length() - 1) + s.substring(0, 0) + ayString;`
e. `pigString = s.substring(1, s.length()) + s.substring(0, 1) + ayString;`

4. This question refers to the `getString` method shown below:

```
public static String getString(String s1, String s2)
{
   int index = s1.indexOf(s2);
   return s1.substring(index, index + s2.length());
}
```

Which is true about `getString`? It may return a string that ...

`I.`　　... is equal to `s2`.
`II.`　　... has no characters in common with `s2`.
`III.`　... is equal to `s1`.

a.　`I` and `III` only　　　d. `I`, `II`, and `III`
b.　`II` and `III` only　　e. None is true
c.　`I` and `II` only