IN-LECTURE QUESTIONS FOR VIDEO 6.5

Note: See the answer key at the bottom of this file.

1. Write the missing code in the line below.

```
Intent ________ = new Intent(Intent.ACTION_SEND)
foo.setType("text/plain");
```

2. Write the missing code to create a simple 'send' intent.

```
Intent i = new Intent(_________)
foo.putExtra(Intent.EXTRA_TEXT,"hello world")
startActivity(foo);
```

3. The following code creates a simple intent. What kind of apps might respond to this intent? Check all that apply.

```
Intent i = new Intent(Intent.ACTION_SEND)
i.setType("text/plain");
i.putExtra(Intent.EXTRA_TEXT,"hello world")
```

   a. Messaging
   b. Email
   c. Bluetooth

4. To send a sms (text) message use:
   a. ACTION_VIEW
   b. ACTION_SEND

5. Write the missing string to set the sms (text) message.

```
intent.setData(Uri.parse("sms:12345678"));
intent.putExtra("_________","My sms text message")
```

6. Check none, one, or both. `Uri.parse("sms:")` and `Uri.fromParts("sms","",null)` are often used with:
   a. setData e.g. `intent.setData(Uri.parse("sms:123456"));`
   b. new Intent constructor. e.g. `new Intent(..., Uri.parse("sms:123456"));`

7. `Uri.parse("sms:123456")` is equivalent to:
   a. `Uri.fromParts("sms","123456",null)`
   b. `Uri.fromParts("sms:","123456",null)`
   c. `Uri.fromParts("sms","123456",true)`

8. Write the missing code. `Uri.fromParts("mailto","",null)` is equivalent to `Uri.parse("_________")`

9. Write the missing code to set the email subject line.

```
Intent i = new Intent(Intent.ACTION_SENDTO);
i.setData(Uri.fromParts("mailto","a@....",null))
i.__________(INTENT.EXTRA_SUBJECT, "Open Immediately - you've won
a iPhone!");
```

10. Write the missing code to set the email text.

```
Intent i = new Intent(Intent.ACTION_SENDTO);
i.setData(Uri.fromParts("mailto","a@....",null))
i.putExtra(INTENT.__________ , "Once upon a time");
```

11. To make your app's use of startActivity (intent) more robust you can:
    a. Call startActivity only if intent.resolveActivity returned a non-null value
    b. Wrap startActivity using a "try-catch" block
    c. Wrap the intent with a chooser intent
    d. All of the other options are correct

12. You can test to see if the user's phone can handle a particular intent by:
    a. Checking if intent is null
    b. Checking if intent.resolveActivity(getPackageManager()) is null
    c. Checking if intent.canRun() is true
    d. Checking if intent.canStart() is true

13. If an intent chooser is used and no activities can handle your underlying intent:
    a. The app crashes and an error message is shown.
    b. The app closes but no error message is shown.
    c. A polite note is displayed to the user.
    d. The user must immediately install a new app.

14. Write the missing code: `if(intent.resolveActivity( getPackageManager())`
    `== ________) { ... }`

15. What does x point to after the following code runs?

```
String x = "Hello";
x=x+" World" + "!" + 123
```

    a. Undefined; this is not valid in Java
    b. A new string object "Hello World!123"
    c. "Hello"
    d. null

16. True or false? Temporary ("Local") variables declared INSIDE one method can be directly accessed from another method.
    a. True; you can use them anywhere in the same Java class.
    b. False; temporary variables can only be used inside their own method.

ANSWER KEY:

1. foo
2. Intent.ACTION_SEND
3. a, b, c
4. a
5. sms_body
6. a, b
7. a
8. mailto:
9. putExtra
10. EXTRA_TEXT
11. d
12. b
13. c
14. null
15. b
16. b