

Northern Michigan University (Marquette Co, MI)

CS345-01-23W Android Programming (Andrew A. Poe) Name: _____
Midterm Page 1/2

Friday 3 March 2023 1:00 P.M. EST

Time: 50 minutes.

1. Write the following method in Kotlin:

`fun HowManyPalindromes (A:Array<String>):Int { ... }`

This method returns the number of palindromes present in array A. For example, if A were {"ANNA","AND","OTTO","DROVE","A","RACECAR"}, the method would return 4. (A palindrome is exactly the same backwards as forwards, like ANNA OTTO A and RACECAR.)

```
fun HowManyPalindromes (A:Array<String>):Int {  
  
    var ct = 0  
    for (s in A)  
        if (s==s.reversed())  
            ct++  
    return ct  
}
```

2. I totally get intents. What I don't get is passing extra parameters using intents. Why can't I just define global variables in my main activity and just have other activities use these global variables?

Sure, it might work, but what might go wrong with this strategy?

Much like using global variables in any time, you run the risk of unintended side effects. Rather than having just two activities have access to this information, the whole program now does which means you either have to designate specific variables for each activity call, which is unwieldy, or you run the risk of overwriting variables some method is going to need.

3. Given the Kotlin class definitions below for a linked list and a linked list node:

```
class LL {  
  
    var head:LLN? = null  
}  
  
class LLN (w:String,n:LLN?) {  
  
    var word = w  
    var next = n  
}
```

Write code for the LL method `fun CountX():Int` . This method returns the number of strings containing the letter X (upper or lowercase) in the linked list. Note that it doesn't count the number of X's themselves. For example, if the list were EXXON-->OWES-->EXTRA -->TAXES, the method should return 3. Even though EXXON has 2 X's, it only counts once.

```
LL:  
fun CountX ():Int {
```

Northern Michigan University (Marquette Co, MI)

CS345-01-23W Android Programming (Andrew A. Poe) Name: _____
Midterm Page 2/2

Friday 3 March 2023 1:00 P.M. EST

```
    if (head == null) return 0
    return head!!.CountX()
}
```

LLN:

```
fun CountX():Int {

    var ct:Int = 0
    if (next != null) ct = next!!.CountX()
    if (word.lowercase().contains ("x"))
        ct++
    return ct
}
```

4. We have a button (myButt) in the main activity of an android app. When I click the button it should run AActivity. After AActivity ends, and my main activity is active again, clicking the same button should cause BActivity to run. Again, after the main activity resumes control, clicking the same button should cause CActivity to run. This time, after the main activity resumes control, clicking the button should do absolutely nothing ever again.

Write the code within these four activities that will make this work.

```
var count=0
myButt.setOnClickListener {

    var i:Intent? = null
    if (count==0) i = AActivity.newIntent(this)
    else if (count==1) i = BActivity.newIntent(this)
    else if (count==2) i = CActivity.newIntent(this)
    if (i!=null) {
        count++
        startActivity (i)
    }
}

AActivity:
fun newIntent (packageContext:Context?):Intent? {

    return Intent (packageContext,AActivity::class.java)
}

BActivity:
fun newIntent (packageContext:Context?):Intent? {

    return Intent (packageContext,BActivity::class.java)
}

CActivity:
fun newIntent (packageContext:Context?):Intent? {
```

Northern Michigan University (Marquette Co, MI)

CS345-01-23W Android Programming (Andrew A. Poe) Name: _____

Midterm Page 3/2

Friday 3 March 2023 1:00 P.M. EST

```
return Intent (packageContext,CActivity: class.java)  
}
```