SENG2200 Week 09 Solutions

1. Difference between pass-by-value and pass-by-reference
   Difference puts a copy of the value in, but there is no other link between the two. Pass by reference passes in an alias to the variable, so any modifications to the param modify the original. For Example:

```
swap(Object a, Object b)
{
        Object temp = b;
        b = a;
        a = temp;
}

main
{
        String one = "abc";
        String two = "def";
        swap(one, two)
        print one;
}
```

   If it is pass by value, it will print "abc", if it is pass by reference it will print "def"
2. What sort of params does Java use?
   Java is pass-by-value. The value of the reference is passed in.
3. What does C++ use?
   C++ is all of them. It can be value, can be reference, can be const-reference

```
void f( int a, int &b, const int &c );
```

   a is value, b is reference, c is const-reference

   4) const cannot be changed. Useful when you want to avoid copying a large item, but don't want it to be changed.

5) 0, 1

6) 7, 1

7)obvious

8) create origin, call origin.distance(point)

9)Any variable that doesn't get modified - so const can be used when it is required (big, nonmodified references).

10) Pass-By-Result uses an 'out parameter' style, overwrites the parameters. I would personally consider this the same as 'reference' where you are overwriting the variable? Pros are variable return types, cons are 'error prone' - exceptions or ordering of params are critical

Pass-By-Value-Result / pass-by-copy

Calling is pass by value, return is pass by result. Same pro's and con's.

11) pass-by-reference can be very efficient for large objects, allows objects to be modified within the function, provides 'full control'. Disadvantages are potentially slower (look ups), and the advanced control allows room for errors.

12) Very hard, open problem (especially in multithreaded environments). Will need to be pass by value, or pass-by-reference with 'copy backs'. Value makes more sense, less overhead.

13) To provide thread safety by ensuring that only one thread executes that section of code at a time.

14) implements runnable, inherits Thread.

runnable allows it to be run as a thread even when the class is already inheriting something.