**iOS Assignment 1**

**Problem 1:**

typedef void (^TestClassCallback)();

When declaring blocks with no parameters, ‘void’ should explicitly be specified to avoid bugs because we can call it in following ways:

void (^TestClassCallback)() = ...

TestClassCallback ();

TestClassCallback (10);

TestClassCallback (@"myString");

This has still not been reflected in old Apple APIs yet maybe because of compatibility reasons. But the correct format should be like:

typedef void (^TestClassCallback)(void);

**Problem 2:**

Since ‘Person’ is a subclass of NSManagedObject, it should never be passed between threads. To pass around a managed object, object ID needs to be passed and then object is retrieved from the context associated to the other thread.

**Problem 3:**

If ‘doworkWithPerson is called 2nd time before the 1st invocation has completed, ‘savedCallback’ will be overwritten and the 1st invocation’s callback will never be called and 2nd invocation’s callback will be called twice. So no need to put a static closure property and I would rather go for a GCD approach like:

- (void)doWorkWithPerson:(Person\*)aPerson callback:(TestClassCallback)aCallback {

dispatch\_async(dispatch\_get\_global\_queue(QOS\_CLASS\_DEFAULT, 0), ^{

[self doVeryLongTask1:aPerson];

aCallback();

});

}

**Problem 4:**

Another problem in the code is: the method ‘doVeryLongTask1’ is being called in background queue and the method is also updating UI (ProgressBar). Therefore, the code that is particular to UI should be called on main queue:

[[ProgressBar instance] update:p];

The UI updates should be called on main queue:

dispatch\_async(dispatch\_get\_main\_queue(), ^{

[[ProgressBar instance] update:p];

});