

LOCATION HANDLING IN IOS

We can easily locate the user's current location in iOS, provided the user allows the application to access the information with the help of the core location framework.

Steps for Location Handling

Step 1 – Create a simple View based application.

Step 2 – Select your project file, then select targets and then add CoreLocation.framework as shown below –

Step 3 – Add two labels in View Controller.Xib and create IBOutlet naming the labels as latitude Label and longitude Label respectively.

Step 4 – Create a new file by selecting File → New → File... → select **Objective C class** and click next.

Step 5 – Name the class as **LocationHandler** with “sub class of” as NSObject.

Step 6 – Select create.

Step 7 – Update **LocationHandler.h** as follows –

```
#import <Foundation/Foundation.h>#import <CoreLocation/CoreLocation.h>
@protocol LocationHandlerDelegate <NSObject>
@required-(void) didUpdateToLocation:(CLLocation*)newLocation
    fromLocation:(CLLocation*)oldLocation;@end
@interface LocationHandler : NSObject<CLLocationManagerDelegate> {
    CLLocationManager *locationManager;}@property(nonatomic,strong)
id<LocationHandlerDelegate> delegate;
+(id)getSharedInstance;-(void)startUpdating;-(void) stopUpdating;
@end
```

Step 8 – Update **LocationHandler.m** as follows –

```
#import "LocationHandler.h"static LocationHandler *DefaultManager = nil;
```

```

@interface LocationHandler()
-(void)initiate;
@end

@implementation LocationHandler
+(id)getInstance{
    if (!DefaultManager) {
        DefaultManager = [[self allocWithZone:NULL]init];
        [DefaultManager initiate];
    }
    return DefaultManager;}

-(void)initiate {
    locationManager = [[CLLocationManager alloc]init];
    locationManager.delegate = self;}

-(void)startUpdating{
    [locationManager startUpdatingLocation];}

-(void) stopUpdating {
    [locationManager stopUpdatingLocation];}

-(void)locationManager:(CLLocationManager *)manager didUpdateToLocation:
(CLLocation *)newLocation fromLocation:(CLLocation *)oldLocation {
    if ([self.delegate respondsToSelector:@selector
(didUpdateToLocation:fromLocation:)]) {
        [self.delegate didUpdateToLocation:oldLocation
        fromLocation:newLocation];
    }}@end

```

Step 9 – Update **ViewController.h** as follows where we have implemented the **LocationHandler delegate** and create two IBOutlet –

```

#import <UIKit/UIKit.h>#import "LocationHandler.h"

@interface ViewController : UIViewController<LocationHandlerDelegate> {
    IBOutlet UILabel *latitudeLabel;
    IBOutlet UILabel *longitudeLabel;}@end

```

Step 10 – Update **ViewController.m** as follows –

```
#import "ViewController.h"

@interface ViewController ()@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];
    [[LocationHandler sharedInstance]setDelegate:self];
    [[LocationHandler sharedInstance]startUpdating];}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.}

- (void)didUpdateToLocation:(CLLocation *)newLocation
fromLocation:(CLLocation *)oldLocation {
    [latitudeLabel setText:[NSString stringWithFormat:
    @"Latitude: %f",newLocation.coordinate.latitude]];
    [longitudeLabel setText:[NSString stringWithFormat:
    @"Longitude: %f",newLocation.coordinate.longitude]];}@end
```

ACCESSING MAPS IN IOS

Maps are always helpful for us to locate places. Maps are integrated in iOS using the MapKit framework.

Steps Involved

Step 1 – Create a simple view-based application.

Step 2 – Select your project file, then select targets and then add MapKit.framework.

Step 3 – We should also add Corelocation.framework.

Step 4 – Add a MapView to ViewController.xib and create an IBOutlet and name it as mapView.

Step 5 – Create a new file by selecting File → New → File... → select Objective C class and click next.

Step 6 – Name the class as MapAnnotation with "sub class of" as NSObject.

Step 7 – Select create.

Step 8 – Update MapAnnotation.h as follows –

```
#import <Foundation/Foundation.h>#import <MapKit/MapKit.h>

@interface MapAnnotation : NSObject<MKAnnotation>@property (nonatomic,
strong) NSString *title;@property (nonatomic, readwrite) CLLocationCoordinate2D
coordinate;

- (id)initWithTitle:(NSString *)title andCoordinate:
    (CLLocationCoordinate2D)coordinate2d;

@end
```

Step 9 – Update **MapAnnotation.m** as follows –

```
#import "MapAnnotation.h"

@implementation MapAnnotation-(id)initWithTitle:(NSString *)title andCoordinate:
```

```

(CLLocationCoordinate2D)coordinate2d {

    self.title = title;
    self.coordinate = coordinate2d;
    return self;}@end

```

Step 10 – Update **ViewController.h** as follows –

```

#import <UIKit/UIKit.h>#import <MapKit/MapKit.h>#import
<CoreLocation/CoreLocation.h>
@interface ViewController : UIViewController<MKMapViewDelegate> {
    MKMapView *mapView;}@end

```

Step 11 – Update **ViewController.m** as follows –

```

#import "ViewController.h"#import "MapAnnotation.h"
@interface ViewController ()@end
@implementation ViewController
- (void)viewDidLoad {
    [super viewDidLoad];
    mapView = [[MKMapView alloc] initWithFrame:
    CGRectMake(10, 100, 300, 300)];
    mapView.delegate = self;
    mapView.centerCoordinate = CLLocationCoordinate2DMake(37.32, -122.03);
    mapView.mapType = MKMapTypeHybrid;
    CLLocationCoordinate2D location;
    location.latitude = (double) 37.332768;
    location.longitude = (double) -122.030039;

    // Add the annotation to our map view
    MapAnnotation *newAnnotation = [[MapAnnotation alloc]
    initWithTitle:@"Apple Head quaters" andCoordinate:location];
    [mapView addAnnotation:newAnnotation];
    CLLocationCoordinate2D location2;

```

```

location2.latitude = (double) 37.35239;
location2.longitude = (double) -122.025919;
MapAnnotation *newAnnotation2 = [[MapAnnotation alloc]
initWithTitle:@"Test annotation" andCoordinate:location2];
[mapView addAnnotation:newAnnotation2];
[self.view addSubview:mapView];}

// When a map annotation point is added, zoom to it (1500 range)-
(void)mapView:(MKMapView *)mv didAddAnnotationViews:(NSArray *)views {
    MKAnnotationView *annotationView = [views objectAtIndex:0];
    id <MKAnnotation> mp = [annotationView annotation];
    MKCoordinateRegion region = MKCoordinateRegionMakeWithDistance
([mp coordinate], 1500, 1500);
    [mv setRegion:region animated:YES];
    [mv selectAnnotation:mp animated:YES];}
- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated. }@end

```