

NAME : NDJANA ESSAGA GERMAIN DAVY
MATRICULE :ICTU1041113

Section A : MCQs

1)A)

2)B

3)C

4)A

5)C

6)A

7)C

8)C

9)A

10)B

11)B

12)D

13)C

14)A

15)B

Section B

1)

Instead of

```
let n1: Int = 1 let n2: Float = 3.0 let n3: Double = 4.65 var result = n1 + n2 + n3
```

we use :

```
var n1 =1
```

```
var n2 :Float
```

```
n2=3.0
```

```
var n3 :Double
```

```
n3=4.65
```

```
var result=n1+n2+n3
```

NAME : NDJANA ESSAGA GERMAIN DAVY
MATRICULE : ICTU1041113

2)

Create the xib and .h/.m files for your custom view.

1a) Assuming you need IBOutlet to view elements over which you want control, give the **File's Owner** the class name defined in your .h file in the identity inspector.

2) In .h file define a property

```
@property (nonatomic, retain) IBOutlet UIView *contentView;
```

2a) define outlets for all the subviews in your xib, to which you want programmatic access.

3) In the .m file synthesize the property and do

```
- (void)awakeFromNib  
{  
    NSLog(@"awake from nib");  
    [[NSBundle mainBundle] loadNibNamed:@"yourNibName" owner:self options:nil];  
    [self addSubview:self.contentView];  
}
```

4) Drag empty views from the palette into their container in the storyboard. Change the class of these views to the class name defined in your .h file.

When you run your app you should see the contents of the xib in your subviews.

5) You can now define outlets to your custom subview instances in the .h file of the container view, and connect them as usual in the storyboard.