(1) How to save Model class in NSUserDefault?
Ans:
E.g. We are having Model class named UserProfile. Now when we want to store object of UserProfile. We will use following code

```
// Storing
    NSData *aDtEncodedObject = [NSKeyedArchiver
archivedDataWithRootObject:aObjUserProfile];
    NSUserDefaults *aUDObj = [NSUserDefaults standardUserDefaults];
    [aUDObj setObject:aDtEncodedObject forKey:key]; // Key will the string which we want
to use as key
    [aUDObj synchronize];

// Accessing
If user wants to access the same object from NSUserDefaults

NSUserDefaults *aUDObj = [NSUserDefaults standardUserDefaults];
  NSData *aDtEncodedObject = [aUDObj objectForKey:key]; // Key will the string which we
used as key

  UserProfile *aObjProfile = [NSKeyedUnarchiver
unarchiveObjectWithData:aDtEncodedObject];
```

(2) Describe view controller lifecycle order when another view controller starts over current view controller (Please explain)
Ans:
>> Require more information as it is confusing. I am assuming that one viewcontroller A is there and another view controller B is a child viewcontroller of viewcontroller A and we are adding Child viewcontroller B in viewwillappear Method of class A. Kindly find the following process of the view controller lifecycle

```
[ClassA loadView]
[ClassA viewDidLoad]
[ClassA viewWillAppear:]
[ClassB loadView]
[ClassB viewDidLoad]
[ClassB viewWillAppear:]
[ClassB viewDidAppear:]
[ClassA viewDidAppear:]
```

(3) How to support multi-resolution and multi-language in iOS app? (Please explain)
Ans:

For Multi Resolution, if design has been given keeping in mind the different resolution then we can apply auto layout and its different property in Storyboard and Xibs. In them we can test for each view and layout including orientation using “Vary of traits” functionality.

For Multi-language,

We will create localizable.string file the project for each language. While we create first time localizable.string file it will ask which files you want to localize. User need to select multiple files for the localization. If there is none then only localizable.string file should be created for each language. It will show in dropdown in file for each language.

User can provide support for different localization by following process from sidemenu

Project >> Info >> Localizations >> Click on “+” button.

User can provide different localize strings with unique key in Localizable.string file and use them in viewcontroller and other classes using following methods

NSLocalizedString(key, comment)

(4) Explain the error handling and exception management

Exception handling

When compiler throws exception at runtime, to handle those exception Apple has provided functionality called @try, @catch and @finally

Using following block we can stop any unwanted or already expected exceptions and provide proper functionality.

Any code which can throw exception must be written in @try block. For handling those exception code must be written in @catch block.

Any code which must be executed, all those code must be written in @finally block

```
@try {
        <#Code that can potentially throw an exception#>
    } @catch (NSException *exception) {
```

```
    NSLog(@"Class:: %s Exception :: %@",__func__,exception.description);
} @finally {

}
```

Error Handling

Any error occurs at runtime must be handled. Apple has provided NSError class to handle errors.

NSError has different properties like domains, code, objects etc.

We can use following kind of code to show error message

```
NSError *error;

If ([error code]){
        NSString *aStrMsg = [error localizedDescription];
        NSString *aStrMoreInfo = [error localizedFailureReason];
}
```