## **Classes**

* how to define class interface and implementation?
* are there constructors and destructors in Objective-C?
* how class is represented at runtime?
* is multiple inheritance allowed?
* what is naming convention for the classes, instances and class members?
* can we mark class as abstract in order to forbid creating instances at compile time?
* what is type introspection?
* what type of the class object? how to get it?
* what functions that perform type introspection do you know? (4)
* how to test class equality of two objects?
* what is the @class directive for? in what cases it can't be used?
* how initializators have to be written?
* how to access to the current object within instance method ?
* how to get instance of class object?
* how can I perform custom class object initialization?
* can I cast class object to some type?
* how to call method of parent class if current class overrides this method?

## **Objects**

* how to create and initialize a class instance ?
* what type initializer should return?
* what steps initializer should perform?
* what are designated and convenience initializers?
* how to send message to the object? with multiple parameters?
* what happens if sending messages to nil? (consider message returns object, pointer, number, struct)
* what dynamic binding means?
* what are steps of message dispatching ?
* can I use the following syntax for message?

[obj performWith:param someAction];

* can we restrict access to the instance methods (make it private) ?
* what is isa variable ?
* can I declare variable like this?

MyClass myVar;

* how to access object instance variables? (NOTE: avoid this, use properties)
* how to refer to ivars within the class and from outside the parent class?

## **Properties**

* what is purpose of properties?
* how to call property using dot syntax and without dot syntax?
* what accessor method names are generated for properties ? (property API contract)
* what are the input parameters and returning values for property accessor methods? (property API contract)
* how to declare property ?
* what do directives @synthesize and @dynamic do ? In which cases they are used?
* what attributes can be used during property declaration? what do they mean? (3 groups of attributes)
* what does the following code snippet mean? What for "= \_varname" is used?

@synthesize prop = \_prop;

* how the following requirements can be implemented: property have to be readonly in public interface,  
  but have to support read/write for internal use ?
* can I override readonly property to be read/write in a descendant class?

## **Protocols**

* what are protocols? what are they for?
* what are formal protocols?
* what are informal protocols?
* how to declare formal protocol? what directives can be used?
* what is default requrement mode?
* how class can adopt the protocol? (what syntax?)
* is multiple protocol adoption allowed?
* how to check that object supports specified protocol?
* how to restrict allowed values for id variable using protocols?
* how to include several protocols into one?
* how conformsToProtocol: works with inherited protocols?
* if protocol has several optional methods and some class adoptes this protocol.  
  How to check that object of this class has implemented specified method?

@protocol P

@optional

-(void)method;

@end

id<P> obj = ....;

[obj method]; // method can be unimplemented, how to check ?

* what is protocol reference?
* what is default modifier: "required" or "optional"?
* how to get protocol object? what it can be used for ?

## **Categories and Extensions**

* what are extensions and categories? what are they for?
* where extensions and categories have to be declared?
* does category allow to add variables to class?
* does category allow to add properties?
* where is extension have to be declared?
* where is category have to be declared? (naming convention)

## **Selectors**

* how to create selector for function with parameters?
* how object of SEL type can be used?
* what is difference between [obj performSelector:@selector(selectorName)] and [obj selectorName] ?
* what will be printed?

@interface A : NSObject

-(SEL)getSelector;

@end

@implementation A

-(SEL)getSelector {

return @selector(getSelector);

}

@end

@interface B : NSObject

-(SEL)getSelector;

@end

@implementation B

-(SEL)getSelector {

return @selector(getSelector);

}

@end

...

A \*a = [[A alloc] init];

B \*b = [[B alloc] init];

NSLog(@"selA == selB: %d", [a getSelector] == [b getSelector]); // ???

NSLog(@"selNew == selA: %d", @selector(getSelector) == [a getSelector]); // ???

## **Associative References**

* what associative references do?
* what is association policy?
* how many associations some object can have?
* how to access associated object?
* how to break association between two objects?

## **Fast Enumerations**

* what syntax is used to iterate a collection using fast enumeration?
* how to make your custom collection to support fast enumeration?
* what standard containers do support fast enumerations?

## **Other stuff**

* what is the difference between #include and #import directives?
* what is id type? Is there type casting warning if you assign to id variable a pointer to another class?

MyClass \*myObj = [[MyClass alloc] init];

id v = myObj;

* what is delegate?
* what are benefits while using statically typed objects instead of id type?