

```
1. //
2. //  Controller.h
3. //  Nib1
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates a Window-based Application implemented with a nib, plus IBAction and IBOutlet.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. @interface Controller : NSObject {
15. }
16.
17. @property (nonatomic, retain) IBOutlet UITextField *textField;
18.
19. - (IBAction)done:(id)sender;
20. - (IBAction)go:(id)sender;
21.
22. @end
```

```
1. //
2. // Controller.m
3. // Nib1
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates a Window-based Application implemented with a nib, plus IBAction and IBOutlet.
10. //
11.
12. #import "Controller.h"
13.
14. @implementation Controller
15.
16. @synthesize textField=_textField;
17.
18. - (IBAction)done:(id)sender
19. {
20.     // hide keyboard
21.     [sender resignFirstResponder];
22. }
23.
24. - (IBAction)go:(id)sender
25. {
26.     // hide keyboard
27.     [self.textField resignFirstResponder];
28.
29.     // show alert
30.     NSString *s = [NSString stringWithFormat:@"Hello, %@", self.textField.text];
31.     UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Hello"
32.                                     message:s
33.                                     delegate:nil
34.                                     cancelButtonTitle:@"Thanks!"
35.                                     otherButtonTitles:nil];
36.     [alert show];
37.     [alert release];
38. }
39.
40. - (void)dealloc
41. {
42.     [_textField release];
43.     [super dealloc];
44. }
45.
46. @end
```

```
1. //
2. //  Nib1AppDelegate.h
3. //  Nib1
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates a Window-based Application implemented with a nib, plus IBAction and IBOutlet.
10. //
11.
12. #import <UIKit/UIKit.h>
13. #import "Controller.h"
14.
15. @interface Nib1AppDelegate : NSObject <UIApplicationDelegate> {
16. }
17.
18. @property (nonatomic, retain) IBOutlet Controller *controller;
19. @property (nonatomic, retain) IBOutlet UIWindow *window;
20.
21. @end
```

```
1. //
2. //  Nib1AppDelegate.m
3. //  Nib1
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates a Window-based Application implemented with a nib, plus IBAction and IBOutlet.
10. //
11.
12. #import "Nib1AppDelegate.h"
13.
14. @implementation Nib1AppDelegate
15.
16. @synthesize controller=_controller;
17. @synthesize window=_window;
18.
19. - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
20. {
21.     [self.window makeKeyAndVisible];
22.     return YES;
23. }
24.
25. - (void)dealloc
26. {
27.     [_controller release];
28.     [_window release];
29.     [super dealloc];
30. }
31.
32. @end
```

```
1. //
2. // Nib1AppDelegate.m
3. // Nib1
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Demonstrates a Window-based Application implemented with a nib, plus IBAction and IBOutlet.
10. //
11.
12. #import <UIKit/UIKit.h>
13.
14. int main(int argc, char *argv[])
15. {
16.     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
17.     int retVal = UIApplicationMain(argc, argv, nil, nil);
18.     [pool release];
19.     return retVal;
20. }
```

```
1. //
2. // Student.h
3. // Students1
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Declares a student.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. @interface Student : NSObject {
15.     @public
16.         int age;
17.         NSString *name;
18. }
19. @end
```

```
1. //
2. // Student.m
3. // Students1
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Defines a student.
10. //
11.
12. #import "Student.h"
13.
14. @implementation Student
15. @end
```

```
1. //
2. //  main.m
3. //  Students1
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates use of a class.
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Student.h"
14.
15. // prototype
16. void greet(Student *s);
17.
18. int main (int argc, const char * argv[])
19. {
20.     // allocate autorelease pool
21.     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
22.
23.     // Alice
24.     Student *alice = [Student alloc];
25.     alice->age = 20;
26.     alice->name = @"Alice";
27.     greet(alice);
28.     [alice release];
29.
30.     // Bob
31.     Student *bob = [Student alloc];
32.     bob->age = 21;
33.     bob->name = @"Bob";
34.     greet(bob);
35.     [bob release];
36.
37.     // drain autorelease pool
38.     [pool drain];
39.
40.     return 0;
41. }
42.
43. // greets student (via stderr)
44. void greet(Student *s)
45. {
46.     NSLog(@"Hello, %@.  I see that are %d years old.\n", s->name, s->age);
47. }
```



```
1. //
2. //  Student.h
3. //  Students2
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Declares a student with getters and setters.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. @interface Student : NSObject {
15.     int _age;
16.     NSString *_name;
17. }
18.
19. - (int)age;
20. - (void)setAge:(int)age;
21.
22. - (NSString *)name;
23. - (void)setName:(NSString *)name;
24.
25. @end
```

```
1. //
2. // Student.m
3. // Students2
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Defines a student with getters and setters.
10. //
11.
12. #import "Student.h"
13.
14. @implementation Student
15.
16. - (int)age
17. {
18.     return _age;
19. }
20.
21. - (void)setAge:(int)age
22. {
23.     _age = age;
24. }
25.
26. - (NSString *)name
27. {
28.     return _name;
29. }
30.
31. - (void)setName:(NSString *)name
32. {
33.     if (_name != name) {
34.         [_name release];
35.         _name = [name copy];
36.     }
37. }
38.
39. - (void)dealloc
40. {
41.     [_name release];
42.     [super dealloc];
43. }
44.
45. @end
```

```
1. //
2. //  main.m
3. //  Students2
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates use of a class, getters, and setters.
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Student.h"
14.
15. // prototype
16. void greet(Student *s);
17.
18. int main (int argc, const char * argv[])
19. {
20.     // allocate autorelease pool
21.     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
22.
23.     // Alice
24.     Student *alice = [Student alloc];
25.     [alice setAge:20];
26.     [alice setName:@"Alice"];
27.     greet(alice);
28.     [alice release];
29.
30.     // Bob
31.     Student *bob = [Student alloc];
32.     [bob setAge:21];
33.     [bob setName:@"Bob"];
34.     greet(bob);
35.     [bob release];
36.
37.     // drain autorelease pool
38.     [pool drain];
39.
40.     return 0;
41. }
42.
43. // greets student (via stderr)
44. void greet(Student *s)
45. {
46.     NSLog(@"Hello, %@.  I see that you are %d years old.\n", [s name], [s age]);
47. }
```

```
1. //
2. //  Student.h
3. //  Students3
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Declares a student with properties.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. @interface Student : NSObject {
15.     int _age;
16.     NSString *_name;
17. }
18.
19. @property (assign, nonatomic, readwrite) int age;
20. @property (copy, nonatomic, readwrite) NSString *name;
21.
22. - (int)age;
23. - (void)setAge:(int)age;
24.
25. - (NSString *)name;
26. - (void)setName:(NSString *)name;
27.
28. @end
```

```
1. //
2. // Student.m
3. // Students3
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Defines a student with properties.
10. //
11.
12. #import "Student.h"
13.
14. @implementation Student
15.
16. - (int)age
17. {
18.     return _age;
19. }
20.
21. - (void)setAge:(int)age
22. {
23.     _age = age;
24. }
25.
26. - (NSString *)name
27. {
28.     return _name;
29. }
30.
31. - (void)setName:(NSString *)name
32. {
33.     if (_name != name)
34.     {
35.         [_name release];
36.         _name = [name copy];
37.     }
38. }
39.
40. - (void)dealloc
41. {
42.     [_name release];
43.     [super dealloc];
44. }
45.
46. @end
```

```
1. //
2. //  main.m
3. //  Students3
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates use of properties.
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Student.h"
14.
15. // prototype
16. void greet(Student *s);
17.
18. int main (int argc, const char * argv[])
19. {
20.     // allocate autorelease pool
21.     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
22.
23.     // Alice
24.     Student *alice = [Student alloc];
25.     alice.age = 20;
26.     alice.name = @"Alice";
27.     greet(alice);
28.     [alice release];
29.
30.     // Bob
31.     Student *bob = [Student alloc];
32.     bob.age = 21;
33.     bob.name = @"Bob";
34.     greet(bob);
35.     [bob release];
36.
37.     // drain autorelease pool
38.     [pool drain];
39.
40.     return 0;
41. }
42.
43. // greets student (via stderr)
44. void greet(Student *s)
45. {
46.     NSLog(@"Hello, %@. I see that you are %d years old.\n", s.name, s.age);
47. }
```

```
1. //
2. //  Student.h
3. //  Students4
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Declares a student with synthesized properties.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. @interface Student : NSObject {
15. }
16.
17. @property (assign, nonatomic, readwrite) int age;
18. @property (copy, nonatomic, readwrite) NSString *name;
19.
20. @end
```

```
1. //
2. //  Student.m
3. //  Students4
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Defines a student with synthesized properties.
10. //
11.
12. #import "Student.h"
13.
14. @implementation Student
15.
16. @synthesize age=_age;
17. @synthesize name=_name;
18.
19. - (void)dealloc
20. {
21.     [_name release];
22.     [super dealloc];
23. }
24.
25. @end
```



```
1. //
2. //  main.m
3. //  Students4
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates use of synthesized properties.
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Student.h"
14.
15. // prototype
16. void greet(Student *s);
17.
18. int main (int argc, const char * argv[])
19. {
20.     // allocate autorelease pool
21.     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
22.
23.     // Alice
24.     Student *alice = [Student alloc];
25.     alice.age = 20;
26.     alice.name = @"Alice";
27.     greet(alice);
28.     [alice release];
29.
30.     // Bob
31.     Student *bob = [Student alloc];
32.     bob.age = 21;
33.     bob.name = @"Bob";
34.     greet(bob);
35.     [bob release];
36.
37.     // drain autorelease pool
38.     [pool drain];
39.
40.     return 0;
41. }
42.
43. // greets student (via stderr)
44. void greet(Student *s)
45. {
46.     NSLog(@"Hello, %@.  I see that you are %d years old.\n", s.name, s.age);
47. }
```

```
1. //
2. //  Student.h
3. //  Students5
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Declares a student with (mostly) synthesized properties.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. @interface Student : NSObject {
15. }
16.
17. @property (assign, nonatomic, readwrite) int age;
18. @property (copy, nonatomic, readwrite) NSString *name;
19.
20. @end
```

```
1. //
2. // Student.m
3. // Students5
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Defines a student with (mostly) synthesized properties.
10. //
11.
12. #import "Student.h"
13.
14. @implementation Student
15.
16. @synthesize age=_age;
17. @synthesize name=_name;
18.
19. - (void)setName:(NSString *)name
20. {
21.     if (_name != name) {
22.         [_name release];
23.         if ([name isEqualToString:@"David"]) {
24.             _name = [[NSString alloc] initWithString:@"Dummy"];
25.         }
26.         else {
27.             _name = [name copy];
28.         }
29.     }
30. }
31.
32. - (void)dealloc
33. {
34.     [_name release];
35.     [super dealloc];
36. }
37.
38. @end
```

```
1. //
2. //  main.m
3. //  Students5
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates use of (mostly) synthesized properties.
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Student.h"
14.
15. // prototype
16. void greet(Student *s);
17.
18. int main (int argc, const char * argv[])
19. {
20.     // allocate autorelease pool
21.     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
22.
23.     // Alice
24.     Student *alice = [Student alloc];
25.     alice.age = 20;
26.     alice.name = @"Alice";
27.     greet(alice);
28.     [alice release];
29.
30.     // Bob
31.     Student *bob = [Student alloc];
32.     bob.age = 21;
33.     bob.name = @"Bob";
34.     greet(bob);
35.     [bob release];
36.
37.     // David
38.     Student *david = [Student alloc];
39.     david.age = 33;
40.     david.name = @"David";
41.     greet(david);
42.     [david release];
43.
44.     // drain autorelease pool
45.     [pool drain];
46.
47.     return 0;
48. }
```

```
49.  
50. // greets student (via stderr)  
51. void greet(Student *s)  
52. {  
53.     NSLog(@"Hello, %@. I see that you are %d years old.\n", s.name, s.age);  
54. }
```

```
1. //
2. //  Student.h
3. //  Students6
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Declares a student with init methods.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. @interface Student : NSObject {
15. }
16.
17. @property (assign, nonatomic, readwrite) int age;
18. @property (copy, nonatomic, readwrite) NSString *name;
19.
20. - (id)initWithName:(NSString *)name andAge:(int)age;
21.
22. @end
```

```
1. //
2. // Student.m
3. // Students6
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Defines a student with init methods.
10. //
11.
12. #import "Student.h"
13.
14. @implementation Student
15.
16. @synthesize age=_age;
17. @synthesize name=_name;
18.
19. - (id)init
20. {
21.     [self initWithName:@"John" andAge:403];
22.     return self;
23. }
24.
25. - (id)initWithName:(NSString *)name andAge:(int)age
26. {
27.     if ((self = [super init]))
28.     {
29.         self.age = age;
30.         self.name = name;
31.     }
32.     return self;
33. }
34.
35. - (void)dealloc
36. {
37.     [_name release];
38.     [super dealloc];
39. }
40.
41. @end
```

```
1. //
2. //  main.m
3. //  Students6
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates init methods.
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Student.h"
14.
15. // prototype
16. void greet(Student *s);
17.
18. int main (int argc, const char * argv[])
19. {
20.     // allocate autorelease pool
21.     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
22.
23.     // Alice
24.     Student *alice = [[Student alloc] initWithName:@"Alice" andAge:20];
25.     greet(alice);
26.     [alice release];
27.
28.     // Bob
29.     Student *bob = [[Student alloc] initWithName:@"Bob" andAge:21];
30.     greet(bob);
31.     [bob release];
32.
33.     // John
34.     Student *john = [[Student alloc] init];
35.     greet(john);
36.     [john release];
37.
38.     // drain autorelease pool
39.     [pool drain];
40.
41.     return 0;
42. }
43.
44. // greets student (via stderr)
45. void greet(Student *s)
46. {
47.     NSLog(@"Hello, %@. I see that you are %d years old.\n", s.name, s.age);
48. }
```



```
1. //
2. //  Student.h
3. //  Students7
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Declares a student with init methods.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. @interface Student : NSObject {
15. }
16.
17. @property (assign, nonatomic, readwrite) int age;
18. @property (copy, nonatomic, readwrite) NSString *name;
19.
20. - (id)initWithName:(NSString *)name andAge:(int)age;
21.
22. @end
```

```
1. //
2. // Student.m
3. // Students7
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Defines a student with init methods.
10. //
11.
12. #import "Student.h"
13.
14. @implementation Student
15.
16. @synthesize age=_age;
17. @synthesize name=_name;
18.
19. - (id)init
20. {
21.     [self initWithName:@"John" andAge:403];
22.     return self;
23. }
24.
25. - (id)initWithName:(NSString *)name andAge:(int)age
26. {
27.     if ((self = [super init]))
28.     {
29.         self.age = age;
30.         self.name = name;
31.     }
32.     return self;
33. }
34.
35. - (void)dealloc
36. {
37.     [_name release];
38.     [super dealloc];
39. }
40.
41. @end
```

```
1. //
2. //  main.m
3. //  Students7
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates mutable arrays.
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Student.h"
14.
15. // prototype
16. void greet(Student *s);
17.
18. int main (int argc, const char * argv[])
19. {
20.     // allocate autorelease pool
21.     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
22.
23.     // allocate array for students
24.     NSMutableArray *students = [[NSMutableArray alloc] init];
25.
26.     // Alice
27.     [students addObject:[Student alloc] initWithName:@"Alice" andAge:20]];
28.
29.     // Bob
30.     [students addObject:[Student alloc] initWithName:@"Bob" andAge:21]];
31.
32.     // greet then release each student
33.     for (id s in students) {
34.         greet(s);
35.         [s release];
36.     }
37.
38.     // release array
39.     [students release];
40.
41.     // drain autorelease pool
42.     [pool drain];
43.
44.     return 0;
45. }
46.
47. // greets student (via stderr)
48. void greet(Student *s)
```

```
49. {  
50.     NSLog(@"Hello, %@. I see that you are %d years old.\n", s.name, s.age);  
51. }
```

```
1. //
2. //  main.m
3. //  Students8
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates immutable arrays.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. int main (int argc, const char * argv[])
15. {
16.     // allocate autorelease pool
17.     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
18.
19.     // allocate array for names
20.     NSArray *names = [NSArray arrayWithObjects:@"Alice", @"Bob", nil];
21.
22.     // greet each student
23.     for (id name in names) {
24.         NSLog(@"Hello, %@.", name);
25.     }
26.
27.     // drain autorelease pool
28.     [pool drain];
29.
30.     return 0;
31. }
```

```
1. //
2. // Student.h
3. // Students9
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Declares a student with convenience method.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. @interface Student : NSObject {
15. }
16.
17. @property (assign, nonatomic, readwrite) int age;
18. @property (copy, nonatomic, readwrite) NSString *name;
19.
20. + (id)studentWithName:(NSString *)name andAge:(int)age;
21. - (id)initWithName:(NSString *)name andAge:(int)age;
22.
23. @end
```

```
1. //
2. // Student.m
3. // Students9
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Defines a student with convenience method.
10. //
11.
12. #import "Student.h"
13.
14. @implementation Student
15.
16. @synthesize age=_age;
17. @synthesize name=_name;
18.
19. + (id)studentWithName:(NSString *)name andAge:(int)age
20. {
21.     return [[[Student alloc] initWithName:name andAge:age] autorelease];
22. }
23.
24. - (id)init
25. {
26.     [self initWithName:@"John" andAge:403];
27.     return self;
28. }
29.
30. - (id)initWithName:(NSString *)name andAge:(int)age
31. {
32.     if ((self = [super init]))
33.     {
34.         self.age = age;
35.         self.name = name;
36.     }
37.     return self;
38. }
39.
40. - (void)dealloc
41. {
42.     [_name release];
43.     [super dealloc];
44. }
45.
46. @end
```

```
1. //
2. //  main.m
3. //  Students9
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates mutable arrays.
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Student.h"
14.
15. int main (int argc, const char * argv[])
16. {
17.     // allocate autorelease pool
18.     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
19.
20.     // allocate array for students
21.     NSMutableArray *students = [NSMutableArray array];
22.
23.     // Alice
24.     Student *alice = [Student studentWithName:@"Alice" andAge:20];
25.     [students addObject:alice];
26.
27.     // Bob
28.     Student *bob = [Student studentWithName:@"Bob" andAge:21];
29.     [students addObject:bob];
30.
31.     // greet each student
32.     for (id student in students) {
33.         NSLog(@"Hello, %@.", [student name]);
34.     }
35.
36.     // drain autorelease pool
37.     [pool drain];
38.
39.     return 0;
40. }
```



```
1. //
2. //  Student.h
3. //  Students10
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Declares a student with (non-init) instance method.
10. //
11.
12. #import <Foundation/Foundation.h>
13.
14. @interface Student : NSObject {
15. }
16.
17. @property (assign, nonatomic, readwrite) int age;
18. @property (copy, nonatomic, readwrite) NSString *name;
19.
20. - (id)initWithName:(NSString *)name andAge:(int)age;
21. - (void)greet;
22.
23. @end
```

```
1. //
2. // Student.m
3. // Students9
4. //
5. // David J. Malan
6. // Harvard University
7. // dmalan@harvard.edu
8. //
9. // Defines a student with (non-init) instance method.
10. //
11.
12. #import "Student.h"
13.
14. @implementation Student
15.
16. @synthesize age=_age;
17. @synthesize name=_name;
18.
19. - (id)init
20. {
21.     [self initWithName:@"John" andAge:403];
22.     return self;
23. }
24.
25. - (id)initWithName:(NSString *)name andAge:(int)age
26. {
27.     if ((self = [super init]))
28.     {
29.         self.age = age;
30.         self.name = name;
31.     }
32.     return self;
33. }
34.
35. - (void)greet
36. {
37.     NSLog(@"Hello, I am %@. I am %d years old.\n", _name, _age);
38. }
39.
40. - (void)dealloc
41. {
42.     [_name release];
43.     [super dealloc];
44. }
45.
46. @end
```

```
1. //
2. //  main.m
3. //  Students10
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Declares a student with (non-init) instance method.
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Student.h"
14.
15. int main (int argc, const char * argv[])
16. {
17.     // allocate autorelease pool
18.     NSAutoreleasePool *pool = [[NSAutoreleasePool alloc] init];
19.
20.     // Alice
21.     Student *alice = [[Student alloc] initWithName:@"Alice" andAge:20];
22.     [alice greet];
23.     [alice release];
24.
25.     // Bob
26.     Student *bob = [[Student alloc] initWithName:@"Bob" andAge:21];
27.     [bob greet];
28.     [bob release];
29.
30.     // drain autorelease pool
31.     [pool drain];
32.
33.     return 0;
34. }
```