

```
1. //
2. // main.m
3. // Students7
4. //
5. // David J. Malan
6. // Harvard University
7. // malan@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.     @autoreleasepool {
21.
22.         // allocate array for students
23.         NSMutableArray *students = [[NSMutableArray alloc] init];
24.
25.         // Alice
26.         Student *alice = [[Student alloc] initWithName:@"Alice" andAge:20];
27.         [students addObject:alice];
28.
29.         // Bob
30.         [students addObject:[[Student alloc] initWithName:@"Bob" andAge:21]];
31.
32.         // greet then release each student
33.         for (Student *s in students) {
34.             greet(s);
35.         }
36.     }
37.     return 0;
38. }
39.
40. // greets student (via stderr)
41. void greet(Student *s)
42. {
43.     NSLog(@"Hello, %@. I see that you are %d years old.\n", s.name, s.age);
44. }
```

```
1. //
2. // Student.h
3. // Students7
4. //
5. // David J. Malan
6. // Harvard University
7. // malan@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. // malan@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 = [self initWithName:@"John" andAge:404];
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. @end
```

```
1. //
2. // main.m
3. // Students7
4. //
5. // David J. Malan
6. // Harvard University
7. // malan@harvard.edu
8. //
9. // Demonstrates mutable arrays.
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Student.h"
14. #import "TF.h"
15.
16. // prototype
17. void greet(Student *s);
18.
19. int main(int argc, const char * argv[])
20. {
21.     @autoreleasepool {
22.
23.         // allocate array for students
24.         NSMutableArray *students = [[NSMutableArray alloc] init];
25.
26.         // Alice
27.         Student *alice = [[Student alloc] initWithName:@"Alice" andAge:20];
28.         [students addObject:alice];
29.
30.         // Bob
31.         [students addObject:[[Student alloc] initWithName:@"Bob" andAge:21]];
32.
33.         TF *chris = [[TF alloc] initWithName:@"Chris" students:students favorite:alice];
34.         NSLog(@"%@", chris.favorite.name);
35.
36.         Student *david = [[Student alloc] init];
37.         david.name = @"David";
38.         david.age = 34;
39.
40.         [students addObject:david];
41.
42.         [chris grade];
43.         [chris outputGrades];
44.
45.         [chris addOfficeHours:[NSDate dateWithNaturalLanguageString:@"2/29/2012"]];
46.         [chris addOfficeHours:[NSDate dateWithNaturalLanguageString:@"3/7/2012"]];
47.         [chris outputOfficeHours];
48.     }
```

```
49.     return 0;
50. }
51.
52. // greets student (via stderr)
53. void greet(Student *s)
54. {
55.     NSLog(@"Hello, %@. I see that you are %d years old.\n", s.name, s.age);
56. }
```

```
1. //
2. // Student.h
3. // Students7
4. //
5. // David J. Malan
6. // Harvard University
7. // malan@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. // malan@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 = [self initWithName:@"John" andAge:404];
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. @end
```

```
1. //
2. // TF.h
3. // Section2
4. //
5. // Created by Christopher Gerber on 2/29/12.
6. // Copyright (c) 2012 Harvard. All rights reserved.
7. //
8.
9. #import <Foundation/Foundation.h>
10.
11. @class Student;
12.
13. @interface TF : NSObject
14.
15. @property (copy, nonatomic) NSString *name;
16. @property (strong, nonatomic, readwrite) NSMutableArray *students;
17. @property (strong, nonatomic, readwrite) NSMutableArray *ohs;
18. @property (strong, nonatomic, readwrite) Student *favorite;
19. @property (strong, nonatomic, readwrite) NSDictionary *grades;
20.
21. - (id)initWithName:(NSString *)newName students:(NSMutableArray *)newStudents favorite:(Student *)newFavorite;
22. - (void)grade;
23. - (void)outputGrades;
24. - (void)addOfficeHours:(NSDate *)date;
25. - (void)outputOfficeHours;
26.
27. @end
```



```
1. //
2. // TF.m
3. // Section2
4. //
5. // Created by Christopher Gerber on 2/29/12.
6. // Copyright (c) 2012 Harvard. All rights reserved.
7. //
8.
9. #import "TF.h"
10. #import "Student.h"
11. #import <stdlib.h>
12.
13. @implementation TF
14.
15. @synthesize name=_name, students=_students, ohs=_ohs, favorite=_favorite, grades=_grades;
16.
17. - (id)initWithName:(NSString *)newName students:(NSMutableArray *)newStudents favorite:(Student *)newFavorite {
18.     if (self = [super init]) {
19.         self.name = newName;
20.         self.students = newStudents;
21.         self.favorite = newFavorite;
22.
23.         self.grades = [[NSMutableDictionary alloc] init];
24.         self.ohs = [[NSMutableArray alloc] init];
25.     }
26.     return self;
27. }
28.
29. - (void)grade {
30.     srand(time(NULL));
31.
32.     for (Student *student in self.students) {
33.         int grade = rand() % 4;
34.         if ([student.name isEqualToString:self.favorite.name])
35.             grade = 4;
36.         [self.grades setValue:[NSNumber numberWithInt:grade] forKey:student.name];
37.     }
38. }
39.
40. - (void)outputGrades {
41.     NSLog(@"Grades:");
42.     for(id student in self.grades)
43.         NSLog(@"%@: %@", student, [self.grades objectForKey:student]);
44. }
45.
46. - (void)addOfficeHours:(NSDate *)date {
47.     [self.ohs addObject:date];
48. }
```

```
49.
50. - (void)outputOfficeHours {
51.     NSLog(@"Office Hours:");
52.
53.     NSDateFormatter *formatter = [[NSDateFormatter alloc] init];
54.     [formatter setDateStyle:NSDateFormatterLongStyle];
55.
56.     for (NSDate *oh in self.ohs)
57.         NSLog(@"%@", [formatter stringFromDate:oh]);
58. }
59. @end
```

```
1. //
2. // main.m
3. // Students7
4. //
5. // David J. Malan
6. // Harvard University
7. // malan@harvard.edu
8. //
9. // Demonstrates mutable arrays.
10. //
11.
12. #import <Foundation/Foundation.h>
13. #import "Student.h"
14. #import "Student+Description.h"
15. #import "TF.h"
16.
17. // prototype
18. void greet(Student *s);
19.
20. int main(int argc, const char * argv[])
21. {
22.     @autoreleasepool {
23.
24.         // allocate array for students
25.         NSMutableArray *students = [[NSMutableArray alloc] init];
26.
27.         // Alice
28.         Student *alice = [[Student alloc] initWithName:@"Alice" andAge:20];
29.         [students addObject:alice];
30.
31.         // Bob
32.         [students addObject:[[Student alloc] initWithName:@"Bob" andAge:21]];
33.
34.         TF *chris = [[TF alloc] initWithName:@"Chris" students:students favorite:alice];
35.         NSLog(@"%@", chris.favorite.name);
36.
37.         Student *david = [[Student alloc] init];
38.         david.name = @"David";
39.         david.age = 34;
40.         NSLog(@"%@", [david description]);
41.
42.         [students addObject:david];
43.
44.         [chris grade];
45.         [chris outputGrades];
46.
47.         [chris addOfficeHours:[NSDate dateWithNaturalLanguageString:@"2/29/2012"]];
48.         [chris addOfficeHours:[NSDate dateWithNaturalLanguageString:@"3/7/2012"]];
```

```
49.     [chris outputOfficeHours];
50.     }
51.     return 0;
52. }
53.
54. // greets student (via stderr)
55. void greet(Student *s)
56. {
57.     NSLog(@"Hello, %@. I see that you are %d years old.\n", s.name, s.age);
58. }
```

```
1. //
2. // Student+Description.h
3. // Section2
4. //
5. // Created by Christopher Gerber on 2/29/12.
6. // Copyright (c) 2012 Harvard. All rights reserved.
7. //
8.
9. #import "Student.h"
10.
11. @interface Student (Description)
12.
13. - (NSString *)description;
14.
15. @end
```

```
1. //
2. // Student+Description.m
3. // Section2
4. //
5. // Created by Christopher Gerber on 2/29/12.
6. // Copyright (c) 2012 Harvard. All rights reserved.
7. //
8.
9. #import "Student+Description.h"
10.
11. @implementation Student (Description)
12.
13. - (NSString *)description {
14.     return [NSString stringWithFormat:@"Name: %@\nAge: %d\n", self.name, self.age];
15. }
16.
17. @end
```

```
1. //
2. // Student.h
3. // Students7
4. //
5. // David J. Malan
6. // Harvard University
7. // malan@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. // malan@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 = [self initWithName:@"John" andAge:404];
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. @end
```



```
1. //
2. // TF.h
3. // Section2
4. //
5. // Created by Christopher Gerber on 2/29/12.
6. // Copyright (c) 2012 Harvard. All rights reserved.
7. //
8.
9. #import <Foundation/Foundation.h>
10.
11. @class Student;
12.
13. @interface TF : NSObject
14.
15. @property (copy, nonatomic) NSString *name;
16. @property (strong, nonatomic, readwrite) NSMutableArray *students;
17. @property (strong, nonatomic, readwrite) NSMutableArray *ohs;
18. @property (strong, nonatomic, readwrite) Student *favorite;
19. @property (strong, nonatomic, readwrite) NSDictionary *grades;
20.
21. - (id)initWithName:(NSString *)newName students:(NSMutableArray *)newStudents favorite:(Student *)newFavorite;
22. - (void)grade;
23. - (void)outputGrades;
24. - (void)addOfficeHours:(NSDate *)date;
25. - (void)outputOfficeHours;
26.
27. @end
```

```
1. //
2. // TF.m
3. // Section2
4. //
5. // Created by Christopher Gerber on 2/29/12.
6. // Copyright (c) 2012 Harvard. All rights reserved.
7. //
8.
9. #import "TF.h"
10. #import "Student.h"
11. #import <stdlib.h>
12.
13. @implementation TF
14.
15. @synthesize name=_name, students=_students, ohs=_ohs, favorite=_favorite, grades=_grades;
16.
17. - (id)initWithName:(NSString *)newName students:(NSMutableArray *)newStudents favorite:(Student *)newFavorite {
18.     if (self = [super init]) {
19.         self.name = newName;
20.         self.students = newStudents;
21.         self.favorite = newFavorite;
22.
23.         self.grades = [[NSMutableDictionary alloc] init];
24.         self.ohs = [[NSMutableArray alloc] init];
25.     }
26.     return self;
27. }
28.
29. - (void)grade {
30.     srand(time(NULL));
31.
32.     for (Student *student in self.students) {
33.         int grade = rand() % 4;
34.         if ([student.name isEqualToString:self.favorite.name])
35.             grade = 4;
36.         [self.grades setValue:[NSNumber numberWithInt:grade] forKey:student.name];
37.     }
38. }
39.
40. - (void)outputGrades {
41.     NSLog(@"Grades:");
42.     for(id student in self.grades)
43.         NSLog(@"%@: %@", student, [self.grades objectForKey:student]);
44. }
45.
46. - (void)addOfficeHours:(NSDate *)date {
47.     [self.ohs addObject:date];
48. }
```

```
49.
50. - (void)outputOfficeHours {
51.     NSLog(@"Office Hours:");
52.
53.     NSDateFormatter *formatter = [[NSDateFormatter alloc] init];
54.     [formatter setDateStyle:NSDateFormatterLongStyle];
55.
56.     for (NSDate *oh in self.ohs)
57.         NSLog(@"%@", [formatter stringFromDate:oh]);
58. }
59. @end
```