

```
1. //
2. //  main.c
3. //  Array
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates arrays.
10. //
11.
12. #include <stdio.h>
13.
14. int main(int argc, const char * argv[])
15. {
16.     // prompt user for number of exams
17.     int n;
18.     printf("Enter number of exams: ");
19.     scanf("%d", &n);
20.
21.     // allocate memory for grades (on stack)
22.     int grades[n];
23.
24.     // prompt user for exams' grades
25.     for (int i = 0; i < n; i++) {
26.         printf("Enter grade %d of %d: ", i+1, n);
27.         scanf("%d", &grades[i]);
28.     }
29.
30.     // do something with grades...
31.
32.     return 0;
33. }
```

```
1. //
2. //  main.c
3. //  Cast
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates explicit casting between primitives.
10. //
11.
12. #include <stdio.h>
13.
14. int main (int argc, const char * argv[])
15. {
16.     char c = 'A';
17.     int i = (int) c;
18.     printf("%c = %d\n", c, i);
19.     return 0;
20. }
```

```
1. //
2. //  main.c
3. //  Conditions
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Reports whether user's input is positive, negative, or zero.
10. //
11.
12. #include <stdio.h>
13.
14. int main(int argc, const char * argv[])
15. {
16.     int n;
17.     printf("Enter an integer: ");
18.     scanf("%d", &n);
19.     if (n > 0) {
20.         printf("Thanks for the positive integer!\n");
21.     }
22.     else if (n < 0) {
23.         printf("Thanks for the negative integer!\n");
24.     }
25.     else {
26.         printf("Thanks for the zero!\n");
27.     }
28.     return 0;
29. }
```

```
1. //
2. //  main.c
3. //  DoWhile
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates a do-while loop.
10. //
11.
12. #include <stdio.h>
13.
14. int main(int argc, const char * argv[])
15. {
16.     int n;
17.     do {
18.         printf("Enter a positive integer: ");
19.         scanf("%d", &n);
20.     }
21.     while (n < 1);
22.     printf("Thanks for the positive integer!\n");
23.     return 0;
24. }
```

```
1. //
2. //  main.c
3. //  Enum
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates enum (and struct).
10. //
11.
12. #include <stdio.h>
13.
14. // gender
15. typedef enum {
16.     FEMALE,
17.     MALE
18. } genders;
19.
20. // student
21. typedef struct {
22.     char *name;
23.     genders gender;
24. } student;
25.
26. // prototype
27. void greet(student s);
28.
29. int main (int argc, const char * argv[])
30. {
31.     // alice
32.     student alice;
33.     alice.name = "Alice";
34.     alice.gender = FEMALE;
35.     greet(alice);
36.
37.     // bob
38.     student bob;
39.     bob.name = "Bob";
40.     bob.gender = MALE;
41.     greet(bob);
42.
43.     return 0;
44. }
45.
46. // greets student
47. void greet(student s)
48. {
```

```
49.     char *title = (s.gender == FEMALE) ? "Ms." : "Mr.";
50.     printf("Hello, %s %s.\n", title, s.name);
51. }
```

```
1. //
2. //  main.c
3. //  For
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates a for loop.
10. //
11.
12. #include <stdio.h>
13.
14. int main(int argc, const char * argv[])
15. {
16.     int n;
17.     printf("Enter a positive integer: ");
18.     scanf("%d", &n);
19.     for (int i = n; i > 0; i--) {
20.         printf("%d...\n", i);
21.     }
22.     printf("Blast off!\n");
23.     return 0;
24. }
```

```
1. //
2. //  main.c
3. //  GetInt
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Gets an int from the user.
10. //
11.
12. #include <stdio.h>
13.
14. int main(int argc, const char * argv[])
15. {
16.     int n;
17.     printf("Enter an integer: ");
18.     scanf("%d", &n);
19.     printf("Thanks for the %d!\n", n);
20.     return 0;
21. }
```



```
1. //
2. //  main.c
3. //  HelloC
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Says hello to the world in C.
10. //
11.
12. #include <stdio.h>
13.
14. int main (int argc, const char * argv[])
15. {
16.     printf("Hello, World!\n");
17.     return 0;
18. }
```

```
1. //
2. //  main.c
3. //  Malloc
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates malloc.
10. //
11.
12. #include <stdio.h>
13. #include <stdlib.h>
14.
15. // prototype
16. int *get_grades(int exams);
17.
18. int main (int argc, const char * argv[])
19. {
20.     // prompt user for number of exams
21.     int n;
22.     printf("Enter number of exams: ");
23.     scanf("%d", &n);
24.
25.     // get grades
26.     int *grades = get_grades(n);
27.
28.     // do something with grades...
29.
30.     // free memory
31.     free(grades);
32.
33.     return 0;
34. }
35.
36. // gets grades
37. int *get_grades(int exams)
38. {
39.     // allocate memory for grades (on heap)
40.     int *grades = malloc(sizeof(int) * exams);
41.
42.     // prompt user for exams' grades
43.     for (int i = 0; i < exams; i++) {
44.         printf("Enter grade %d of %d: ", i+1, exams);
45.         scanf("%d", &grades[i]);
46.     }
47.     return grades;
48. }
```

```
1. //
2. //  main.c
3. //  SizeOfPrimitives
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Prints some pointers' sizes.
10. //
11.
12. #include <stdio.h>
13.
14. int main(int argc, const char * argv[])
15. {
16.     printf("double: %lu\n", sizeof(double *));
17.     printf("float: %lu\n", sizeof(float *));
18.     printf("int: %lu\n", sizeof(int *));
19.     printf("long long: %lu\n", sizeof(long long *));
20.     return 0;
21. }
```

```
1. //
2. //  main.c
3. //  SizeOfPrimitives
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Prints some primitives' sizes.
10. //
11.
12. #include <stdio.h>
13.
14. int main(int argc, const char * argv[])
15. {
16.     printf("double: %lu\n", sizeof(double));
17.     printf("float: %lu\n", sizeof(float));
18.     printf("int: %lu\n", sizeof(int));
19.     printf("long long: %lu\n", sizeof(long long));
20.     return 0;
21. }
```

```
1. //
2. //  main.c
3. //  Struct
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates struct.
10. //
11.
12. #include <stdio.h>
13.
14. // student
15. typedef struct {
16.     int age;
17.     char *name;
18. } student;
19.
20. // prototype
21. void greet(student s);
22.
23. int main (int argc, const char * argv[])
24. {
25.     // alice
26.     student alice;
27.     alice.age = 20;
28.     alice.name = "Alice";
29.     greet(alice);
30.
31.     // bob
32.     student bob;
33.     bob.age = 21;
34.     bob.name = "Bob";
35.     greet(bob);
36.
37.     return 0;
38. }
39.
40. // greets student
41. void greet(student s)
42. {
43.     printf("Hello, %s. I see that you are %d years old.\n", s.name, s.age);
44. }
```

```
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.
15. @interface Student : NSObject {
16. @public
17.     int age;
18.     NSString *name;
19. }
20. @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.c
3. //  SwapFailure
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Fails to swap two variables' values.
10. //
11.
12. #include <stdio.h>
13.
14. // function prototype
15. void swap(int a, int b);
16.
17.
18. int main(int argc, const char * argv[])
19. {
20.     int x = 0;
21.     int y = 1;
22.
23.     printf("x is %d\n", x);
24.     printf("y is %d\n", y);
25.     printf("Swapping x and y...\n");
26.     swap(x, y);
27.     printf("Success!\n");
28.     printf("x is %d\n", x);
29.     printf("y is %d\n", y);
30.
31.     return 0;
32. }
33.
34.
35. //
36. //  Swaps arguments' values.
37. //
38.
39. void swap(int a, int b)
40. {
41.     int tmp = a;
42.     a = b;
43.     b = tmp;
44. }
```



```
1. //
2. //  main.c
3. //  SwapSuccess
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Swaps two variables' values.
10. //
11.
12. #include <stdio.h>
13.
14. // function prototype
15. void swap(int *a, int *b);
16.
17.
18. int main(int argc, const char * argv[])
19. {
20.     int x = 0;
21.     int y = 1;
22.
23.     printf("x is %d\n", x);
24.     printf("y is %d\n", y);
25.     printf("Swapping x and y...\n");
26.     swap(&x, &y);
27.     printf("Success!\n");
28.     printf("x is %d\n", x);
29.     printf("y is %d\n", y);
30.
31.     return 0;
32. }
33.
34.
35. //
36. //  Swaps arguments' values.
37. //
38.
39. void swap(int *a, int *b)
40. {
41.     int tmp = *a;
42.     *a = *b;
43.     *b = tmp;
44. }
```

```
1. //
2. //  main.c
3. //  While
4. //
5. //  David J. Malan
6. //  Harvard University
7. //  dmalan@harvard.edu
8. //
9. //  Demonstrates a while loop.
10. //
11.
12. #include <stdio.h>
13.
14. int main(int argc, const char * argv[])
15. {
16.     int n;
17.     printf("Enter a positive integer: ");
18.     scanf("%d", &n);
19.     while (n > 0) {
20.         printf("%d...\n", n);
21.         n--;
22.     }
23.     printf("Blast off!\n");
24.     return 0;
25. }
```