

```

1. Array
2. Array[3]
3. &Array[3]
4. *(&Array[3])
5. int Array[10];
6. int x;
7. int* px;
8. &x
9. *px
10. x

```

Illustration 1: Object Declarations and References

1. Illustration 1: Which line declares an array?
 - A. 1
 - B. 2
 - C. 4
 - D. 5
2. Illustration 1: Which line declares an integer?
 - A. 7
 - B. 5
 - C. 6
 - D. 10
3. Illustration 1: Which line declares an integer pointer?
 - A. 8
 - B. 9
 - C. 7
 - D. 10
4. Illustration 1: Which line gives the value of the fourth element of an array?
 - A. 2
 - B. 3
 - C. 1
 - D. 5
5. Illustration 1: Which line gives the address of the third subscripted value of an array?
 - A. 8
 - B. 3
 - C. 2
 - D. 4
6. Illustration 1: Which line specifies the value of an integer?
 - A. 8
 - B. 10
 - C. 7
 - D. 1
7. Illustration 1: Which line gives the pointer to the first element of an array?
 - A. 1
 - B. 5
 - C. 7
 - D. 9
8. Illustration 1: Which line gives the contents of the address of the fourth element of an array?
 - A. 2
 - B. 3
 - C. 4
 - D. 1
9. Illustration 1: Which line gives the address of an integer?
 - A. 7
 - B. 10
 - C. 8
 - D. 9
10. Which line dereferences (gives the contents) of an address given by an integer pointer?
 - A. 6
 - B. 10
 - C. 9
 - D. 8

```

1. int x;
2. int cherry[10];
3. foo(x);
4. foo(&x);
5. foo(cherry[3]);
6. foo(cherry, 3);
7. foo(&cherry[3]);

```

Illustration 2: Function calls – Ch 5.4

11. Illustration 2: Which line declares an integer?
 - A. 1
 - B. 2
 - C. 3
 - D. 6
12. Illustration 2: Which line declares an array with 10 elements?
 - A. 3
 - B. 5
 - C. 2
 - D. 7
13. Illustration 2: Which line calls foo with a copy of an integer?
 - A. 4
 - B. 6
 - C. 7
 - D. 3
14. Illustration 2: Which line calls foo with the address of an integer?
 - A. 4
 - B. 3
 - C. 6
 - D. 7
15. Illustration 2: Which line calls foo with the address of the first element of an array?
 - A. 4
 - B. 5
 - C. 6
 - D. 7

16. Illustration 2: Which line calls foo with the value of an element of an array?

- A. 4
- B. 5
- C. 6
- D. 7

17. Illustration 2: Which line calls foo with the address of an element of an array which is not the first element?

- A. 6
- B. 7
- C. 5
- D. 4

```
1. int foo( int boot[5] );
2. int foo( int* boot );
3. int foo( int boot[] );
4. int foo ( int x );
5. int foo( int *x );
6. int foo( int tire[3][5] );
```

Illustration 3: Function prototypes – Ch 5.6

18. Illustration 3: What makes the lines a function prototype?

- A. int
- B. foo
- C. parenthesis
- D. semicolon

19. Illustration 3: Which line declares a single dimension integer array with an unnecessary dimension value?

- A. 3
- B. 6
- C. 1
- D. 5

20. Illustration 3: Which line declares a multi-dimension array?

- A. 6
- B. 1
- C. 3
- D. 5

21. Illustration 3: Which line declares an integer array without including dimension information?

- A. 1
- B. 5
- C. 3
- D. 6

22. Illustration 3: Which line declares that the function is receiving a copy of a value?

- A. 1
- B. 4
- C. 2
- D. 5

23. Illustration 3: Does line 1 declare an integer pointer?

- A. no
- B. yes
- C. 5
- D. '*'

24. Illustration 3: Does line 2 declare an integer pointer?

- A. yes
- B. '*'
- C. int
- D. no

25. Illustration 3: Does line 3 declare an integer pointer?

- A. yes
- B. []
- C. no
- D. int

26. Illustration 3: Can the integer pointer from line 2 be used with subscripts?

- A. No
- B. boot
- C. yes
- D. only outside function

27. Illustration 3: Can the integer pointer from line 3 be used with subscripts?

- A. Yes
- B. {}
- C. boot
- D. no

28. Illustration 3: Which line above declares an integer value?

- A. 4
- B. 5
- C. 1
- D. 6

29. Illustration 3: Could line 5 be used to declare an array?

- A. no
- B. yes
- C. with []
- D. int

30. Illustration 3: Could line 5 be used to declare the pointer to a single integer value?

- A. no
- B. as an array
- C. ()
- D. yes

31. Illustration 3: Which line is most useful to declare a reference to (the pointer to) a specific element of an integer array (such as &donut[4]) ?

- A. 1
- B. 3
- C. 5
- D. 2

32. Illustration 3: Which line is most useful to declare the value of a specific element of an integer array (such as donut[4]) ?

- A. 1
- B. 6
- C. 4
- D. 3

- ```

1. int foo(tube[])
2. int foo(int x)
3. int foo(int* px)
4. int foo(int* tube, size)

```

*Illustration 4: Function definitions Ch 5.5*

33. Illustration 4: Which function definition lines can be called with the name of an array?
- 1,3,4
  - 1,3
  - 3,4
  - 4
34. Illustration 4: Which function definition lines can be called with a copy of the original value?
- 1
  - 4
  - 3
  - 2
35. Illustration 4: Which function definition can be called with an array value (like `foo(array[3]);`)?
- 2
  - 1
  - 4
  - 3
36. Illustration 4: Which function definition can include the dimension property of an array?
- 1
  - 3
  - 4
  - 2
37. Illustration 4: Which function definition can be called with the address of an integer value?
- 3
  - 2
  - 4
  - 1

- ```

1. int boot [10];
2. int x;
3. y = foo(x);
4. y = foo(boot[5]);
5. y = foo(&x);
6. y = foo(boot);
7. y= foo(boot, 10);

```

Illustration 5: Function calls – Ch 7.4

39. Illustration 5: Which line calls a function with a pointer to an array?
- 6
 - 2
 - 4
 - 5
40. Illustration 5: Which line calls a function with dimension property of an array passed as an integer?
- 7
 - 6
 - 4
 - 1
41. Illustration 5: Which line calls a function with a copy of an array element?
- 1
 - 3
 - 4
 - 6
42. Illustration 5: Which line calls a function with the address of an integer?
- 5
 - 6
 - 3
 - 7
43. Illustration 5: Which line calls a function with the copy of an integer value?
- 2
 - 6
 - 4
 - 3
44. String is a data type in C?
- true
 - false
 - only as class
 - instances
45. Strings are letters placed in character arrays ending with a null character
- integers
 - classes
 - false
 - true
46. Illustration 6: Which line declares a string (char array) with maximum 80 characters, initialized to a string?
- 1
 - 2
 - 4
 - 6
47. Illustration 6: Which line just creates an array useful for 80 characters?
- 4
 - 6
 - 5
 - 1
38. Illustration 5: Which line calls a function with the name of an array?
- 6
 - 2
 - 4
 - 3

```

1.char sta[80];
2.char text[80]="This is the value";
3.char* ptext="Sample string";
4.char output[120];
5.puts(text);
6.printf("%s",text);

```

Illustration 6: Ch 8

48. Consider Illustration 6 which line outputs string array **'text'** with a new line included?
- 2
 - 5
 - 4
 - 6
49. Consider Illustration 6 line 5, what does the calling argument supply to the function?
- int
 - value
 - pointer
 - 80
50. Consider Illustration 6 which line uses the string format specifier?
- 3
 - 6
 - 5
 - 1
51. Consider Illustration 6 line 6, what does the second calling argument supply to the function?
- int
 - value
 - pointer
 - 80
52. Consider Illustration 6 line 6, what does the first calling argument supply to the function?
- Int
 - value
 - pointer
 - 80
53. Consider Illustration 6 Which line creates a string of letters and stores them in array **'text'**?
- 2
 - 1
 - 3
 - 5
54. Consider Illustration 6 line 3 creates a string constant and stores what in ptext?
- value
 - text
 - pointer
 - 80
55. Illustration 6: Which string library function can move **'text'** to **'output'**?
- strcmp
 - strlen
 - streat
 - strcpy
56. Illustration 6: Which string library function can add **'ptext'** to **'output'**?
- streat
 - strcpy
 - strlen
 - strcmp
57. Illustration 6: Which string library function can add "this is more text" to **'output'**?
- strcmp
 - strcpy
 - streat
 - strlen
58. Illustration 6: Which string library function can tell how many characters are in **'text'**?
- strcpy
 - strlen
 - streat
 - strcmp
59. Illustration 6: Which string library function can tell if 'text' contains "This is the value"?
- strlen
 - strcpy
 - strcmp
 - streat
60. Illustration 6: Which string contains 14 characters?
- 1
 - 2
 - 3
 - 4