

# Academic Challenge



EASTERN ILLINOIS UNIVERSITY

## 2019 Academic Challenge

### COMPUTER SCIENCE TEST – STATE

Computer Science Test Production Team

Kevin Schaefer, Independent Contractor – Author

#### GENERAL DIRECTIONS

Please read the following instructions carefully. This is a timed test; any instructions from the test supervisor should be followed promptly.

The test supervisor will give instructions for filling in any necessary information on the answer sheet. Most Academic Challenge sites will ask you to indicate your answer to each question by marking an oval that corresponds to the correct answer for that question. One oval should be marked to answer each question. Multiple ovals will automatically be graded as an incorrect answer.

Be sure ovals are marked as  , not  ,  ,  , etc.

If you wish to change an answer, erase your first mark completely before marking your new choice.

You are advised to use your time effectively and to work as rapidly as you can without losing accuracy. Do not waste your time on questions that seem too difficult for you. Go on to the other questions, and then come back to the difficult ones later if time remains.

**Time: 40 Minutes**

**Number of Questions: 30**

**DO NOT OPEN TEST BOOKLET UNTIL YOU ARE TOLD TO DO SO!**

©2019 Eastern Illinois University

All rights reserved



Academic Challenge  
2019 State Computer Science Exam

1. Given that a computer uses Two's Complement notation to store decimal integers, what range of decimal values can be represented, given that 10 bits are used to store numbers?
  - A. -511 to 511
  - B. -512 to 511
  - C. -511 to 512
  - D. -256 to 255
  - E. -255 to 256
  
2. Which of the following statements is true regarding Quicksort and Heapsort?
  - A. Heapsort is a stable sorting algorithm
  - B. Heapsort and Mergesort can both be used as searching algorithms
  - C. Mergesort can be implemented as a stable sorting algorithm
  - D. Heapsort has a worst-case Big-Oh time complexity of  $O(n^2)$
  - E. Both C and D
  
3. Suppose two classes are defined, `Tree` and `Pine`, with this relationship: `class Pine : private Tree`. If an instance of `Pine` is instantiated, how many times does a constructor in `Tree` get called?
  - A. 0 times, since instantiating an instance of `Pine` will only call a constructor on `Pine`, but not any base classes
  - B. 0 times, since `Tree` is privately inherited by `Pine`
  - C. 1 time
  - D. There is not enough information given
  - E. Both A and B.
  
4. Which of the following statements is true regarding sorting algorithms?
  - A. Stable sorting algorithms maintain the relative position of equal elements in the list
  - B. The time complexity of a sorting algorithm describes how quickly a sorting algorithm can sort an input list
  - C. Only stable sorting algorithms work on large data sets
  - D. Both A and B
  - E. Both A and C

5. Suppose a WiFi router is operating on the IEEE 802.11G standard, and is capable of transmitting data at 54 mbps. If the client is 10m away from the router, what is the theoretical maximum number of bits that the router could transmit before the first bit reaches the client? Assume the speed of light is  $3.0 \times 10^8$  m/s.
- A. 1.8 bits
  - B. 54 megabits
  - C.  $(10/3) \times 10^{-8}$  bits
  - D. 64 kilobits
  - E. 180 bits

6. Given the following routing table, which interface would be used to send a packet if the destination address is 10.17.0.5?

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
default	10.36.0.1	0.0.0.0	UG	0	0	0	wlan0
10.0.0.0	*	255.255.255.0	U	0	0	0	br-lan
10.36.0.0	*	255.255.0.0	U	0	0	0	wlan0
10.36.0.1	*	255.255.255.255	UH	0	0	0	wlan0

- A. wlan0
  - B. br-lan
  - C. 10.36.0.1
  - D. 255.0.0.0
  - E. 127.0.0.1
7. Which of the following statements is true regarding the usage of template classes and template functions?
- A. Using template classes/functions is an example of polymorphism
  - B. The Standard Template Library (STL) is composed of template classes
  - C. Template classes cannot be used with inheritance
  - D. Both A and B are correct
  - E. Answers A, B, and C are correct
8. In C++, what is the purpose of `return 0;` at the end of `int main()`?
- A. Indicates that the program executed successfully; non-zero return codes indicate a non-successful exit
  - B. A leftover relic of the C language
  - C. A required line of code at the end of `int main`, which is used to know when the program should terminate
  - D. All of the above
  - E. Both B and C

Use the following code for questions 9, 10, and 11:

```

1  int size = 3;
2  int a[size][size], i, j;
3
4  for(i = 0; i < size; i++)
5  {
6      a[0][i] = 1;
7      a[i][0] = 1;
8  }
9
10 for(i = 1; i < size; i++)
11 {
12     for(j = 1; j < size; j++)
13     {
14         a[i][j] = a[i - 1][j] + a[i][j - 1];
15     }
16 }
17
18 for(i = 0; i < size; i++)
19 {
20     for(j = 0; j < size; j++)
21     {
22         cout << setw(4) << a[i][j];
23     }
24
25     cout << endl;
26 }
```

9. What is the purpose of `setw(4)` on line 22?
- Sets floating-point rounding to 4 decimal places
  - Pads the next value to be printed with 4 zeros on the left
  - Pads the next value to be printed with 4 spaces on the left
  - Pads the previously printed value with 4 zeros on the right
  - Pads the previously printed value with 4 spaces on the right
10. What are the contents of the array `a` after the code executes?
- { { 1, 2, 3 }, { 2, 3, 4 }, { 3, 4, 5 } }
  - { { 1, 1, 1 }, { 1, 2, 3 }, { 1, 3, 6 } }
  - { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } }
  - { { 1, 1, 1 }, { 2, 2, 2 }, { 3, 3, 3 } }
  - None of the above

11. Which of the following is a valid way to access element `a[2][1]`?
- A. `*(a[2] + 1)`
  - B. `*(a[1] + 2)`
  - C. `*a[2] + 1`
  - D. `&(a[2] + 1)`
  - E. None of the above
12. Convert the value 11011100 from Two's Complement to One's Complement.
- A. 00100100
  - B. 11011011
  - C. 11011100
  - D. 00100011
  - E. 11011101

Use the following code for questions 13, 14, 15, and 16:

```
1 int fun2(int &a, int b)
2 {
3     return a = a - b;
4 }
5
6 int fun1(int &a, int b)
7 {
8     if(a % b)
9         return fun1(--a, b);
10    else
11        return fun2(a, b);
12 }
```

13. Under which of the following circumstances will `fun1` call `fun2`?
- A. `a` is divisible by `b`
  - B. `a` is not divisible by `b`
  - C. `fun2` will never be called from `fun1`
  - D. `fun2` will always be called from `fun1`
  - E. None of the above
14. What is the result of the function call `fun1(3, 2);`?
- A. 0
  - B. 1
  - C. 2
  - D. 3
  - E. Compile-time exception

15. What is the result of the function call `int a = 3; fun1(a, 2);`?
- A. 0
  - B. 1
  - C. 2
  - D. 3
  - E. Compile-time exception
16. How many times is `fun1` called with the function call `int a = 3; fun1(a, 2);`?
- A. 0
  - B. 1
  - C. 2
  - D. 3
  - E. 4

Use the following code for questions 17, 18, and 19:

```
1  int i = 3;
2
3  for(; i--; --i) i--;
```

17. What value does the variable `i` contain after the code executes?
- A. -1
  - B. 0
  - C. 5
  - D. The code results in an infinite loop
  - E. The code does not compile
18. How many times is the conditional in the `for`-loop executed?
- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. There is not enough information
19. If the `for`-loop were changed to the following, what would the result be?
- ```
for(; --i; --i);
```
- A. -2
  - B. -1
  - C. 0
  - D. The code results in an infinite loop
  - E. The code does not compile

20. Which of the following is equivalent to  $a = b / 2$ ;, assuming both `a` and `b` are `ints` and are non-negative?
- A. `a /= b`;
  - B. `a = b >> 2`;
  - C. `a = b & 2`;
  - D. `a = b << 2`;
  - E. None of the above
21. Which of the following are benefits of Public-Key Encryption?
- A. Compression
  - B. Encryption
  - C. Authentication
  - D. None of the above
  - E. Both B and C
22. Which of the following statements is true regarding min-heaps?
- A. The heap is a complete binary tree
  - B. The data at any node is less than or equal to the data in that node's parent
  - C. The data at any node is less than or equal to the data in that node's children
  - D. The data at any node is less than or equal to the data in that node's right sub-tree
  - E. Both A and C
23. Is it possible to define a template struct in C++?
- A. Yes, as long as the template type is a primitive type
  - B. Yes
  - C. No, only template classes or template functions can be defined
  - D. No, since structs can't have private members
  - E. None of the above
24. Suppose two classes are defined: `class Class1` and `class Class2 : public Class1`. Is it possible for `Class2` to have both a Has-A relationship and an Is-A relationship with `Class1`?
- A. No, a class can only have one type of relationship with another class
  - B. No, if a class inherits from a base class, it can't also have an instance of that base class as a property of the class
  - C. Yes, as long as `Class2` publicly inherits from `Class1`
  - D. Yes, it is possible for `Class2` to have both relationship types with `Class1`
  - E. None of the above



25. Suppose an algorithm takes  $2n^2 + n^4 + 500n$  steps. What is this expressed in Big-Oh notation?

- A.  $O(2n^2 + n^4)$
- B.  $O(500n)$
- C.  $O(2n^2)$
- D.  $O(n^4)$
- E.  $O(n)$

Use the following code for question 26:

```
1 struct node
2 {
3     node *left;
4     int data;
5     node *right;
6 };
```

26. Which of the following data structures could be created using the `node` struct?

- A. Binary tree
- B. Max heap
- C. Linked List
- D. All of the above
- E. None of the above

27. Which of the following statements is true regarding Hexadecimal numbers?

- A. A single Hex digit can be represented with 3 binary bits
- B. The largest single-digit Hex value is F
- C. Hexadecimal can be converted to Octal by shifting the bits to the right by one space
- D. Both A and B
- E. Both B and C

Use the following code for questions 28, 29, and 30:

```

1  #include <iostream>
2  using namespace std;
3
4  struct Color
5  {
6      short red, green, blue;
7  };
8
9  template <class T>
10 class IconBase
11 {
12     public:
13         virtual void SetColorAt(int x, int y, T color) = 0;
14         virtual T     GetColorAt(int x, int y)         = 0;
15 };
16
17 class GrayscaleIcon : public IconBase<short>
18 {
19     protected:
20         short data[16][16];
21     public:
22         void SetColorAt(int x, int y, short color);
23         short GetColorAt(int x, int y) { return this->data[x][y]; };
24 };
25
26 void GrayscaleIcon::SetColorAt(int x, int y, short color)
27 {
28     this->data[x][y] = color > 255 ? 255 : color;
29 }
30
31 class Icon : public IconBase<Color>
32 {
33     private:
34         GrayscaleIcon red;
35         GrayscaleIcon blue;
36         GrayscaleIcon green;
37     public:
38         void SetColorAt(int x, int y, Color color);
39         Color GetColorAt(int x, int y);
40 };
41
42 void Icon::SetColorAt(int x, int y, Color color)
43 {
44     this->red.SetColorAt(x, y, color.red);
45     this->blue.SetColorAt(x, y, color.blue);
46     this->green.SetColorAt(x, y, color.green);
47 }

```

```
48 Color Icon::GetColorAt(int x, int y)
49 {
50     Color tmp;
51     tmp.red = this->red.GetColorAt(x, y);
52     tmp.blue = this->blue.GetColorAt(x, y);
53     tmp.green = this->green.GetColorAt(x, y);
54     return tmp;
55 }
56
57 short ConvertToGrayscale(Color c)
58 {
59     return (c.red + c.green + c.blue) / 3;
60 }
61
62 int main()
63 {
64     return 0;
65 }
```

28. What is the relationship between `Icon::SetColorAt` and `GrayscaleIcon::SetColorAt`?
- A. Data-hiding
  - B. Overloading
  - C. Inheritance
  - D. Is-A
  - E. None of the above
29. What does the `ConvertToGrayscale` function do?
- A. Returns an object of type `GrayscaleIcon`
  - B. Throws an exception if the red, green, or blue component values are zero
  - C. Return the max of the red, green, and blue components
  - D. Return the average value of the red, green, and blue components
  - E. None of the above
30. Would it make sense to declare the private variable `data` on `GrayscaleIcon` as `const`?
- A. Yes, as it would prevent `data` from accidentally being reassigned to a new array
  - B. Yes, because of the Principal of Least Privilege
  - C. No, as doing so would prevent it from legitimately being updated
  - D. No, as arrays cannot be declared as `const`
  - E. Both C and D

