



## 2014 Academic Challenge

### COMPUTER SCIENCE TEST - STATE

This Test Consists of 30 Questions

#### Computer Science Test Production Team

Jim Feher, McKendree University – Author/Team Leader  
Nathan White, McKendree University – Author  
Scott Elliott, John A. Logan College – Reviewer  
Mary Weaver, WYSE – Coordinator of Test Production

#### 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. Only one oval should be marked to answer each question. Multiple ovals will automatically be graded as incorrect answers.

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 \*\*\***

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

© 2014 Worldwide Youth in Science and Engineering

“WYSE”, “Worldwide Youth in Science and Engineering” and the “WYSE Design” are service marks of and this work is the Copyright © 2014 of the Board of Trustees of the University of Illinois at Urbana - Champaign.

All rights reserved

WYSE – Academic Challenge  
Computer Science Test (State) – 2014

1. Authentication for a computer system can best be described as which of the following?
  - a. Encrypting data so that others cannot view the data being exchanged.
  - b. Digitally signing the information so that others can verify who it came from.
  - c. Compressing the data so that it can be sent more efficiently.
  - d. Verification of the identity of the user.
  - e. Adding extra information in case the data transmitted is damaged during transmission.
  
2. Which of the following statements is true regarding cache memory?
  - a. Cache allows large programs to be run by dividing the program into parts and storing some of the parts on the hard drive.
  - b. Cache has information stored in it by the manufacturer and is not volatile.
  - c. Cache improves processing by acting as a temporary high-speed holding area between RAM and the CPU.
  - d. Cache is similar to both RAM and ROM—it can be updated and does not lose the information when power is removed.
  - e. Cache cannot be updated and loses all information when power is removed.
  
3. What does a system bus do?
  - a. Provides a socket for external devices to connect to the system unit.
  - b. Connects the CPU to memory on the motherboard.
  - c. Drives inputs from the CPU to the output devices.
  - d. Converts AC power to DC power.
  - e. Provides a holding area for data and instructions.
  
4. Which of the following is true about a disk defragmenter?
  - a. It locates and eliminates unnecessary fragments.
  - b. It rearranges files and unused disk space to optimize operations.
  - c. One is provided with the Windows operating system.
  - d. a, b, and c are all true.
  - e. None of the above are true.
  
5. Convert the floating point hexadecimal number,  $A3.8$ , into the equivalent floating point decimal.
  - a. 163.8
  - b. 10100011.0111
  - c. 168.5
  - d. 163.25
  - e. None of the above.
  
6. What is the decimal value of the base 7 number 64?
  - a. 100000
  - b. 64
  - c. 47
  - d. 46
  - e. 98

Use the following code for Questions 0, 8, 9, & 10.

```

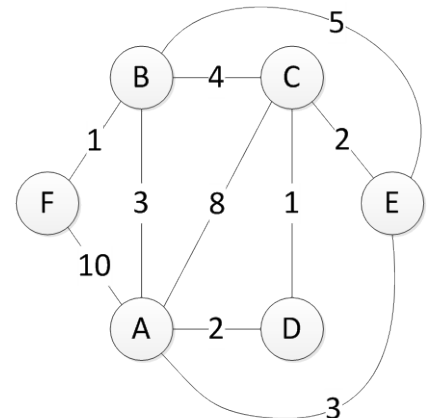
1   class Rectangle {
2       public:
3           Rectangle(float=1, float=1);
4           float getLength()    const { return length;          }
5           float getWidth ()    const { return width;          }
6           float getPerimeter () const { return 2*width+2*length; }
7           float getArea()      const { return width*length;   }
8           bool  setWidth (float);
9           bool  setLength(float);
10          protected:
11              float length, width;
12          };
13          Rectangle::Rectangle(float l, float w) {
14              setLength(l);
15              setWidth(w);
16          } // end Rectangle constructor
17          bool Rectangle::setLength(float l) {
18              length = (l > 0) ? l : 2;
19              return (l > 0); // if l was valid, return true
20          } // end setLength method
21          bool Rectangle::setWidth(float w) {
22              width = (w > 0) ? w : 2;
23              return (w > 0); // if w was valid, return true
24          } // end setWidth method

25          class Box: public Rectangle {
26              public:
27                  Box(float=1, float=1, float=1);
28                  float getHeight()    const { return height;          }
29                  float getVolume ()    const { return width*length*height; }
30                  float getPerimeter() const { return 4*(width+length+height); }
31                  float getArea()      const;
32                  bool  setHeight (float);
33                  Box operator++();
34              private:
35                  float height;
36          };
37          Box::Box (float l, float w, float h) {
38              setLength(l);
39              setWidth(w);
40              setHeight(h);
41          }; // end Box constructor
42          float Box::getArea() const {
43              return 2 * (width*length + width*height + height*length);
44          } // end getArea method
45          bool Box::setHeight(float h) {
46              height = (h > 0) ? h : 2;
47              return (h > 0); // if h was valid, return true
48          } // end setHeight method
49          Box Box::operator++ () {
50              width++; height++; length++;
51              return *this;
52          } // end overloaded ++

```

7. The code from lines 49 to 52 is best described as which of the following?
- addition operator
  - pre-increment
  - post-increment
  - assignment
  - post-decrement
8. The `getArea` method on line 42 is best described by which of the following?
- constructor
  - abstraction
  - overriding
  - inheritance
  - inline
9. If `protected` on line 10 were changed to `private`, list the methods in `Box` that would now not work.
- All methods including the overloaded `++`.
  - `getPerimeter`, `getArea`, `getHeight`, `getVolume`
  - `getPerimeter`, `getArea`, `getHeight`, `getVolume` and `++`
  - `getPerimeter`, `getArea`, `getVolume` and `++`
  - `getPerimeter`, `getArea`, `getHeight`, `setHeight`, `getVolume` and `++`
10. For the following objects instantiated, how many constructors are called?
- ```
Rectangle a, b(2, -3),
Box c[3];
```
- 3
  - 4
  - 5
  - 7
  - 8

11. The graph to the right has weights associated with each of its edges. Which of the following is not true regarding a minimal spanning tree for the graph?
- It includes the edges BF, AD, CD and CE.
  - It includes a total of 5 edges.
  - It includes the edges BF, AB, CD and AD.
  - It will not include the edge AF.
  - It includes the loop AD, DC, CE, EA.



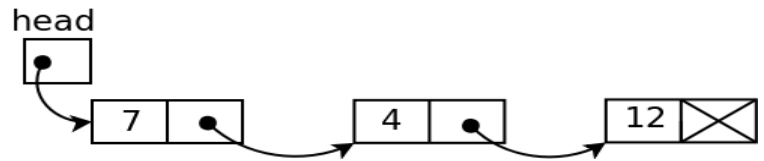
12. Which of the following functional operations would not cause a list to change?
- bubble sort
  - binary search
  - merge sort
  - quick sort
  - All cause the list to change.

13. Which code segment will create a new node with a data element of 3 and put it directly after the 7 in the list below?

```

1  typedef struct node* nodeRef;
2  struct node {
3      int data;
4      nodeRef next;
5  };
6
7  nodeRef head = new node, curNode;
8

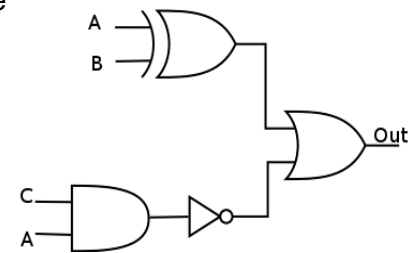
```



- a. `curNode = new node;`  
`curNode->data = 3;`  
`curNode->next = head->next;`  
`head->next = curNode;`
- b. `curNode = new node;`  
`head->next = curNode;`  
`curNode->data = 3;`  
`curNode->next = head->next;`
- c. `curNode = new node;`  
`curNode.data = 3;`  
`curNode.next = head->next;`  
`head.next = curNode;`
- d. `curNode = new node;`  
`curNode = 3;`  
`curNode->next = head->next;`  
`head->next = curNode;`
- e. None of the segments work properly.

14. What is the equivalent representation of the following circuit if A and C are both true?

- a. TRUE
- b. FALSE
- c. B
- d. B'
- e. (A xor B) and not (C and A)



15. Simplify the following Karnaugh map to produce the minimal sum of products expression.

- a. C'D' or B'D' or A'CD'
- b. C'D' or CD'
- c. A'D' or C'D'
- d. C'D' or A'CD' or AB'CD'
- e. C'D' or B'D' or A'D'

|      | A'B' | A'B | AB | AB' |
|------|------|-----|----|-----|
| C'D' | T    | T   | T  | T   |
| C'D  | F    | F   | F  | F   |
| CD   | F    | F   | F  | F   |
| CD'  | T    | T   | F  | T   |

16. Rate the following functions in terms of rate of growth from slowest to fastest as n increases.

- a.  $n, n^2, \log_2 n, 2^n, n!$
- b.  $n, \log_2 n, n^2, 2^n, n!$
- c.  $\log_2 n, n, n^2, 2^n, n!$
- d.  $\log_2 n, n, 2^n, n^2, n!$
- e.  $\log_2 n, n, n^2, n!, 2^n$

17. Which of the following statements is true regarding dynamically allocated memory?
- Once it is created, it cannot be deallocated until the program is complete.
  - It is allocated when the program is compiled.
  - It is allocated when the program is executing.
  - It is allocated when the program is started.
  - It cannot be allocated during a function call.

Use the following code for Questions 18 & 19.

```

1  int myfunc(int xx, int yy) {
2      static int zz = 0;
3      for (int i = 0; i < yy; i++)
4          zz += xx;
5      return zz;
6  }
7  int main()
8  {
9      int z=0;
10     z = myfunc(4,2);
11     cout << z << endl;
12     z = myfunc(2,2);
13     cout << z << endl;
14     return 0;
15 }
```

18. For printing output from cout, stdout is the standard output that a program uses. What will be output to stdout?
- |      |      |       |      |       |
|------|------|-------|------|-------|
| a. 8 | b. 8 | c. 12 | d. 4 | e. 12 |
| 4    | 12   | 8     | 8    | 16    |
19. If static was removed from line 2, what would be output to stdout?
- |      |      |       |      |       |
|------|------|-------|------|-------|
| a. 8 | b. 8 | c. 12 | d. 4 | e. 12 |
| 4    | 12   | 8     | 8    | 16    |
20. What is a set of overloaded functions?
- A set of functions that contain no function calls.
  - A series of functions that have function prototypes.
  - A series of functions that have the same name that are differentiated by their parameter lists.
  - A set of functions that calls each other.
  - None of the above.

Use the following code for Questions 21, 22, & 23.

```

1   int myfunc(int xx, int yy) {
2       if (yy)
3           return xx*myfunc(xx, yy-1);
4       else if (yy == 0)
5           return 1;
6       else
7           exit(1);
8   }
9   int main()
10  {
11      int z;
12      z = myfunc(4.0,5.0);
13      cout << z << endl;
14      return 0;
15  }
```

21. The call to myfunc on line 3 is commonly referred to as what?
- A special procedure call.
  - A system call.
  - It will result in a runtime error for the values provided.
  - A recursive call.
  - A nuisance call.
22. What will be output to stdout?
- 4
  - 16
  - 32
  - 728
  - 1024
23. What triggers the base case in myfunc?
- yy == 0
  - yy > 1
  - yy != 0
  - yy == xx
  - xx == 1

The rest of this page is left blank intentionally.

Use the following code for Questions 24, 25, 26, & 27.

```

1   int main()
2   {
3       int x(0);
4       char ch('y');
5       try {
6           while (ch=='y' || ch=='Y') {
7               cout << "Enter a number: ";
8               cin >> x;
9               try {
10                  if (x>20)
11                      throw "Number too big";
12                  if (x<10)
13                      throw x;
14                  cout << "Good Number";
15                  cout << "Do you want to continue? ";
16                  cin >> ch;
17              }
18              catch (const char MyMsg[]) {
19                  cout << "Program ending";
20              }
21          }
22      }
23      catch (int z) {
24          cout << "Program is stopping";
25      }
26      catch (...) {
27          cout << "Program is over";
28      }
29      return 0;
30  }
```

24. What user input will cause line 19 to execute?
- a. 4                      b. 12                      c. 16                      d. 20                      e. 24
25. After line 19 executes, what will be the next thing output to stdout?
- a. Enter a number:  
b. Program ending  
c. Program is stopping  
d. Program is over  
e. Nothing further will be written to stdout.
26. What user input will cause line 24 to execute?
- a. 4                      b. 12                      c. 16                      d. 20                      e. 24



27. What is the meaning of the ellipses on line 26?
- The catch block will handle any exceptions where the throw operand is an integer.
  - The catch block will handle any exceptions where the throw operand is a character array.
  - The catch block will handle any exception thrown in the try block that has not already been handled by a previous catch block.
  - The catch block will handle any exceptions where the throw operand is a literal.
  - None of the above.

Use the following code for Questions 28, 29, & 30.

```

1 void myfunc(char* str) {
2     int i = 0;
3     int j = 0;
4     while ((*str+i) = *(str+j++)) != '\0')
5         if (*(str+i) != ' ')
6             i++;
7     return;
8 }
9 int main()
10 {
11     char str[]="this is my string";
12     cout << str << endl;
13     myfunc(str);
14     cout << str << endl;
15     return 0;
16 }
```

28. What will be output to stdout from line 14?
- this is my string
  - THIS IS MY STRING
  - thisismystring
  - this
  - string
29. What does the '\0' represent on line 4?
- Zero
  - The Null character
  - A carriage return/line feed
  - An uppercase A
  - A lowercase a
30. What will be output to stdout from line 14 if `str` was initialized to "string"?
- string
  - STRING
  - string\0
  - STRING\0
  - None of the above