

University Interscholastic League

# **Computer Science Competition 2004-05**

**SAC PRACTICE TEST 1**

- Remember that calculators are NOT permitted.

**QUESTION 1**

What is the number  $81_{10}$  when converted to binary?

- A. 1010010    B. 1011010    C. 1100000    D. 1001001    E. None of these

**QUESTION 2**

Which of the following declares variable `a` to be of type `Address`, and initializes it to the address below?

1600 Pennsylvania Avenue  
Washington, DC 20500

- A. `Address a = new Address("1600", "Pennsylvania Avenue", "Washington", "DC", "20500");`
- B. `Address a = new Address(1600, "Pennsylvania Avenue", "Washington", "DC", 20500);`
- C. `Address a = Address("1600", "Pennsylvania Avenue", "Washington", "DC", "20500");`
- D. `Address a = Address(1600, "Pennsylvania Avenue", "Washington", "DC", 20500);`
- E. None of these

```
public class Address {
    public Address (String hn,
                   String strt,
                   String ci,
                   String stat,
                   String zip) {
        // code to initialize private
        // data members
    }

    // other methods not shown

    private String houseNumber;
    private String street;
    private String city;
    private String state;
    private String zipCode;
}
```

**QUESTION 3**

Which of these expressions could be used in the constructor to test that parameter `stat` is exactly two characters long?

- A. `stat == 2`
- B. `stat.charAt(2) == null`
- C. `stat.length == 2`
- D. `stat.length() == 2`
- E. More than one of these

**QUESTION 4**

What replaces `<*1>` in the code to the right to indicate that class `C` is a subclass of class `B`?

- A. subclass of class B
- B. subclass of B
- C. `: public B`
- D. extends B
- E. None of these

```
public class C <*1> {
    // methods and data not shown
}
```

**QUESTION 5**

Which sorting algorithm is being implemented by the code to the right?

- A. Selection
- B. Insertion
- C. Mergesort
- D. Quicksort
- E. None of these

```
public static void sort(int[] a) {
    for (int i=0; i < a.length; ++i) {
        int min=a[i], minIndex=i;
        for (int j=i+1; j < a.length; ++j)
            if (a[j]<a[minIndex]) {
                min=a[j]; minIndex=j;
            }
        a[minIndex]=a[i];
        a[i]=min;
    }
}
```

**QUESTION 6**

What is returned by `mystery(15, 3)`?

- A. 0
- B. 1
- C. 2
- D. 3
- E. None of these

```
public static int mystery(int x, int y) {
    if (x%y == 0) return y;
    else return mystery(y, x%y);
}
```

**QUESTION 7**

What is returned by `mystery(48, 80)`?

- A. 8
- B. 48
- C. 16
- D. 80
- E. None of these

**QUESTION 8**

What is output by the code to the right if `int i` is 3?

- A. Nothing
- B. \*\*
- C. \*\*\*
- D. \*\*\*\*
- E. None of these

```
for (int j=i; j>0; --j)
    System.out.print('*');
```

**QUESTION 9**

What is output by the code to the right if `int i` is -2?

- A. Nothing
- B. \*
- C. More than 5 \*'s
- D. \*\*
- E. None of these

<p><b>QUESTION 10</b></p> <p>What is output by the code to the right?</p> <p>A. Nothing  B. UIL Computer Science  C. Compute  D. Comp  E. None of these</p>	<pre>String s = "UIL Computer Science"; System.out.print(s.substring(4,7));</pre>
<p><b>QUESTION 11</b></p> <p>What is output by the code below?</p> <pre>A a = new A(5); System.out.print(a.f());</pre> <p>A. 0                                    B. 1  C. 3                                    D. 5  E. None of these</p>	<pre>public class A {     public A(int x) {         this.x = x;     }     public int f() {         return x - 3;     }     private int x; }  public class B extends A {     public B(int x, int y) {         super(x);         this.y = y;     }     public int f() {         return y + 5 + super.f();     }     private int y; }</pre>
<p><b>QUESTION 12</b></p> <p>What is output by the code below?</p> <pre>A a = new B(5,10); System.out.print(a.f());</pre> <p>A. 15                                    B. 16  C. 17                                    D. 18  E. None of these</p>	<pre>int n; // code to initialize n  int count = 0;  for (int i=0; i&lt;n; ++i)     for (int j=0; j&lt;n; ++j)         count++;</pre>
<p><b>QUESTION 13</b></p> <p>If <code>int n</code> is initialized to 10, what is the value of <code>count</code> after executing the code to the right?</p> <p>A. 45                                    B. 81  C. 55                                    D. 100  E. None of these</p>	<pre>int n; // code to initialize n  int count = 0;  for (int i=0; i&lt;n; ++i)     for (int j=0; j&lt;n; ++j)         count++;</pre>
<p><b>QUESTION 14</b></p> <p>What is the running time of the nested loop in the code to the right? Give the smallest correct answer.</p> <p>A. <math>O(1)</math>                                    B. <math>O(n)</math>  C. <math>O(n^2)</math>                                    D. <math>O(n^3)</math>  E. None of these</p>	

**QUESTION 15**

What is the value of `roster[2]` after the declaration below?

```
Employee[] roster = new Employee[100];
```

- A. null
- B. undefined
- C. 0
- D. invalid declaration
- E. None of these

```
public class Employee {
    public Employee(String f, String l,
                    long id, int b, long s) {
        firstName = f;
        lastName = l;
        IDNumber = id;
        band = b;
        salary = s;
    }

    public void raise() {
        salary += salary*band/100;
    }

    public double getSalary() {
        return salary;
    }

    // other methods not shown

    private String firstName, lastName;
    private long IDNumber;
    private int band;
    private long salary;
}
```

**QUESTION 16**

Assume part of `roster` is initialized. Which of these gives raises to all employees, ignoring empty slots?

- A. 

```
for (int i=0; i<100; ++i)
    roster[i].raise();
```
- B. 

```
roster.raise();
```
- C. 

```
for (int i=0; i<100; ++i)
    if (roster[i].firstName)
        roster[i].raise();
```
- D. 

```
for (int i=0; i<100; ++i)
    if (roster[i])
        roster[i].raise();
```
- E. None of these

**QUESTION 17**

What is output by the code to the right on the input below?

```
hello
1
```

- A. h
- B. hello
- C. e
- D. hello1
- E. None of these

```
// Assume getString() and getInt() are
// static methods in a class named IO that
// read a String and an integer from
// the keyboard
```

```
String s = IO.getString();
int i = IO.getInt();

try {
    System.out.print(s.charAt(i));
}
catch (StringIndexOutOfBoundsException e) {
    System.out.print("Invalid: " + i);
}
```

**QUESTION 18**

What is output by the code to the right on the input below?

```
hello
15
```

- A. h
- B. Invalid: 15
- C. e
- D. nothing
- E. None of these

**QUESTION 19**

Which of the following is not a subclass of Object?

- A. String    B. TreeSet    C. Comparable    D. Integer    E. None of these

**QUESTION 20**

What does `int[] intArray` look like after the static method call `process(intArray)` when `intArray` is the array below?

1	2	3	4	5	6
---	---	---	---	---	---

A. 

1	2	3	4	5	6
---	---	---	---	---	---

B. 

6	5	4	3	2	1
---	---	---	---	---	---

C. 

1	2	3	3	2	1
---	---	---	---	---	---

D. 

6	5	4	4	5	6
---	---	---	---	---	---

- E. None of these

```
public static void process(int[] A) {  
    for (int i=0; i<A.length/2; ++i)  
        A[A.length - i - 1] = A[i];  
}
```

# Computer Science Answer Key

## UIL Practice 2004-05

1. E - The correct answer is 1010001 ( $64 + 16 + 1$ )
2. A - Two things are being tested, needing to use keyword `new` to build an object, and seeing that the parameters to the constructor are all strings, so they all need to be surrounded by double quotes
3. D - This is what the `length()` method does. Answer C would be correct for an array, not a string.
4. D - Tests syntax of inheritance
5. A - Choosing the element which goes in each spot from smallest to largest
6. D - The code implements Euclid's greatest common divisor algorithm, enabling a fast solution for those who recognize the algorithm. (The `%` operator is remainder.)
7. C - Frequently there will be an easy case and a hard case on a recursion problem, where to solve the hard case there's not enough time to follow the recursion through, but instead you have to figure out what's being computed.
8. C - Normal workings of a for loop with a counter decreasing from 3 to 0.
9. A - The "test" happens first, so code inside the loop is not always executed.
10. E - Correct answer "Com". String indices start at 0. The `substring()` method gives substring between first index inclusive and last index exclusive, so in this case characters 4, 5, and 6.
11. E - Correct answer 2. A simple method call.
12. C - Method determination is by object type, not reference type, so the call `a.f()` calls the version of `f()` from class B.
13. D - 10 inner loop iterations for each of 10 outer loop iterations.
14. C - Running time proportional to  $n^2$ .
15. A - Initializing an array of references does not create object for those references, but sets them all to `null`.
16. E - The correct test in the `if` statement is `roster[i] != null`
17. C - `charAt()` gets a character at the specified index.
18. B - Standard behavior of a `try/catch` block. The `charAt()` method throws the exception caught.
19. C - `Comparable` is an interface, not a class.
20. C - Each iteration of the loop replaces item `(length-n)` with item `n`.