

Version: I

Leave this grid blank, please.

I	II	III	Total

### PLEASE READ THESE INSTRUCTIONS:

- (i) **Do not start yet! Keep the exam closed until the official start.**
- (ii) One two-sided sheet of self-prepared notes only is allowed; otherwise, **no books, notes, or calculating devices**. You are **not** allowed to use your own scratch paper. The back pages of this exam may be used as such. Should you need more paper during the exam, more is available at the front of the room.
- (iii) Keep your photo-ID out on your desktop where the proctors can find it. They will come by during the test to check it.
- (iv) There is a **strict** time limit of **1 hour and 50 minutes** for this exam. Time warnings will occasionally be posted on the board in front.
- (v) This exam consists of three parts. There are 19 multiple-choice questions worth 2 points each in Part I, 3 short-answer questions worth 5 points each in Part II, and 2 short coding questions worth 25 points each in Part III. The total number of points possible is 103, of which 3 points are extra credit. The questions are **not** necessarily in order of increasing difficulty! Answer the easier questions first. Save difficult questions for last.
- (vi) Should corrections or hints be deemed necessary, they will be written on the board in front.
- (vii) Please provide the information requested below. Scores will be posted online at the `my.ucla` web site in a couple of days.

Name (print) \_\_\_\_\_

E-mail address: \_\_\_\_\_

Student ID number: \_\_\_\_\_

Signature: \_\_\_\_\_

**Part I: Multiple choice questions (2 points each)**

Select the single best answer to each question.

Indicate your answers in the following table by **CROSSING THEM OUT**.

Example:

0	a	×	c	d	e
---	---	---	---	---	---

1	a	b	c	d	e	f
2	a	b	c	d	e	f
3	a	b	c	d	e	f
4	a	b	c	d	e	f
5	a	b	c	d	e	f
6	a	b	c	d	e	f
7	a	b	c	d	e	f
8	a	b	c	d	e	f
9	a	b	c	d	e	f
10	a	b	c	d	e	f

11	a	b	c	d	e	f
12	a	b	c	d	e	f
13	a	b	c	d	e	f
14	a	b	c	d	e	f
15	a	b	c	d	e	f
16	a	b	c	d	e	f
17	a	b	c	d	e	f
18	a	b	c	d	e	f
19	a	b	c	d	e	f

**Question 1.** How does the compiler match a function call to a function definition?

- (a) By the name of the function only.
- (b) By the name of the function and return type only.
- (c) By the name of the function, the names and types of input arguments, and the name and type of the object or value returned.
- (d) By the name of the function and the number and types of its arguments.
- (e) Compilers don't have to match function calls to function definitions.

**Question 2.** The difference between an argument received by value and an argument received by reference is that

- (a) a value argument is just another name for an existing storage location.
- (b) a reference argument is just another name for an existing storage location.
- (c) a value argument shares its storage with the calling module.
- (d) a reference argument requires local storage separate from any storage used by the calling module.
- (e) a value argument is really just the address of a storage location.

**Question 3.** The first step in debugging is

- (a) recrimination (b) globalization (c) localization (d) recursion (e) induction

**Question 4.** Which form of source-code solution is better?

- (a) one big self-contained function  
(b) a collection of small, simple, interacting functions  
(c) It depends on the problem being solved.  
(d) It depends on whether the program is written by one person or by several people.  
(e) It depends on whether the programming language used supports breaking a program into subprograms.

**Question 5.** What is the output of the following C++ fragment?

```
char a[5] = {'a', 'b', 'c'};
for (int i = 1; i <= 3; ++i )
    cout << a[i] << " ";
```

- (a) a b c  
(b) b c a  
(c) b c 0  
(d) b c  
(e) The output is unpredictable.

**Question 6.** Which of the following operators is associated from right to left?

- (a) + (b) && (c) = (d) == (e) . (the dot operator)

**Question 7.** When called, what does the following function print out?

```
void foo(){ int i; std::cout << i; }
```

- (a) 0 (b) 1 (c) This function will not compile. (d) -INT\_MAX (e) The output is in general unpredictable.

**Question 8.** When does the following function return true?

```
bool goo( int i ){ return !(i%4) ; }
```

- (a) When  $i == 0$   
(b) When  $i != 0$   
(c) When  $i$  is divisible by 4  
(d) When  $i$  is NOT divisible by 4  
(e) The return value cannot be predicted, even when  $i$  is known.

**Question 9.** How many different integer values can be represented with 32 bits ?

- (a)  $2^8$  (b)  $2^{16} - 1$  (c)  $2^{8-1} = 2^7$  (d)  $2^{32}$  (e)  $2^{64} - 1$

**Question 10.** How many steps are needed by an efficient algorithm to determine whether a given value  $x$  is included in a *sorted* list of  $N$  values?

- (a)  $2^N$  (b)  $2N$  (c)  $N/2$  (d)  $\log_2 N$  (e)  $N! = N(N-1)(N-2)\cdots 1$ .

**Question 11.** Which statement is most accurate?

- (a) Global named constants are bad, but global variables are good.  
 (b) Global named constants are good, but global variables are bad.  
 (c) Global named constants are good, and global variables are good.  
 (d) Global named constants are bad, and global variables are bad.  
 (e) Global constants and variables are no better or worse than local variables and constants.

**Question 12.** Suppose A and B are boolean expressions. Which of the following expressions is (or are) always true?

- (a)  $!(A \ || \ B) == !A \ || \ !B$  (b)  $!(A \ || \ B) == !A \ \&\& \ !B$  (c)  $!(A \ \&\& \ B) == !A \ \&\& \ !B$   
 (d)  $!(A \ \&\& \ B) == !A \ || \ !B$  (e) (a) and (c) (f) (b) and (d)

**Matching.** *Select your answers to the next six questions from the following list of definitions.*

- (a) A problem-solving technique whereby a problem is repeatedly reduced to successively smaller instances of itself until easy-to-solve base cases are reached.  
 (b) A set of rules for assigning bit sequences to values.  
 (c) A chunk of storage.  
 (d) A graduated system of simplified models progressing from superficial and global to detailed and local.  
 (e) The hierarchical organization of loosely coupled components, each of which communicates with the others through a prescribed interface.  
 (f) A sequence of symbols reducible to a single value of a specific type.

**Question 13.** What is an expression? **Question 14.** What is recursion?

**Question 15.** What is an object? **Question 16.** What is modularity?

**Question 17.** What are levels of abstraction? **Question 18.** What is type?

**Question 19.** Given representable `int` values `a` and `b`, which of the following code fragments correctly evaluates to `true` if and only if `a+b` is representable as type `int`?

- (a) `a + b <= INT_MAX` (b) `a + b >= INT_MAX` (c) `b <= INT_MAX - a`  
 (d) `a >= INT_MAX - b` (e) none of the preceding

**Part II: Short Answer Questions (5 points each)**

**Question 20.** What does the following C++ function do?

```
int puzzle(char a[]){
    int ans=0, i=0;
    while (a[i]){
        assert ( '0' <= a[i] && a[i] <= '9' );
        ans = 10*ans + (a[i++]-'0');
    }
    return ans;
}
```

**Question 21.** Given the definitions

```
int i = 15, j = 6, k;
double x = 4.0, y = 3.0, z;
```

evaluate the following expressions.

(a)  $i / j$       (b)  $i \% j$       (c)  $k = i / x$       (d)  $z = i / x$       (e)  $z = i / j$

**Question 22.** List the most important element(s) of good coding style and explain briefly (1–3 sentences) why they are so important.

<b>Part III: Coding (25 points each)</b>
--

*Partial credit is awarded to partial solutions, including pseudocode. Style and efficiency count, but detailed documentation is not expected. State any simplifying assumptions that you make. You may define as many auxiliary functions as you wish to support your answer. Your solution to one question may call your solutions to other questions freely.*

**Question 23.**

- (a) **(15 points)** Implement a C++ function `int scoreCount(int a[], int n, int s)` which, given an array  $a$  of  $n$  integer scores and an integer  $s$ , determines how many of the first  $n$  elements of  $a$  are equal to  $s$ . The function returns this total to the caller. For example, given the following data stored in integer array  $a$  ...

85 75 90 82 85 75 90 82 67 75 82 64 82

... the value of the expression `scoreCount( a, 13, 75 )` is 3.

- (b) **(10 points)** Implement a C++ function `int mode(int a[], int n)` which, given an array  $a$  of  $n$  integer scores, determines and returns the value of the most frequently occurring score in  $a$ . For example, given the same data listed in the example for the previous question, the value of `mode( a, 13 )` is 82.

**Question 24.**

- (a) **(15 points)** Implement a C++ function `int substringStart( char phrase[], char text[] )` to determine the location, if any, of the first occurrence of a given phrase in a given text. The phrase is said to occur in the text if its elements are stored contiguously in the text in the same order in which they appear in the phrase. E.g., phrase “blue” occurs in the text “big blue sky” at position 4, but it does *not* occur in the text “boil mud eggs.” Both phrase and text are represented as C-style arrays. If the phrase does not occur in the text, the function returns `-1`; otherwise, it returns the integer index of the position in the text of the first letter of the phrase (4 in the “big blue sky” example above).
- (b) **(10 points)** Implement a C++ function `int phraseCount( char phrase[], char text[] )` which returns the number of occurrences of a given phrase in a given text, phrase and text both stored as C-style arrays. For example, the phrase “blue” appears twice in the text “blue dog blue sky.” *Hint. You can reuse your solution to the preceding problem.*

**Scratch Paper**

**Scratch Paper**

**Scratch Paper**