

Name: _____

Username: _____

- Section:** 07 - Sushant Athley, Tuesday 11:30am
(check one) 08 - Aishwarya Bhide, Thursday 11:30am
 09 - Phanindra Kumar, Tuesday 2:30pm
 10 - Phanindra Kumar, Thursday 2:30pm
 16 - Sudip Mittal, Tuesday 10:00am
 17 - Sushant Athley, Thursday 4:00pm

	Score	Max
I.		20
II. p. 3		16
p. 4		16
p. 5		8
III. p. 6		10
p. 7		10
p. 8		10
p. 9		10
Total		100

Instructions:

1. This is a closed-book, closed-notes exam.
2. You have 75 minutes for the exam.
3. Calculators, cell phones and laptops must be put away.
4. Clearly indicate your final answer.

I. Multiple Choice (2 points each)

For each question in this section, circle **ONE** answer. Choose the **BEST** answer.

1. C++ allows functions to be overloaded. This means
 - (a) You can have two functions with the same name in the same scope.
 - (b) You can call a function with more arguments than it has formal parameters.
 - (c) You can call the function multiple times in parallel.
 - (d) A function is allowed to call itself as long as its name is in the same namespace.

2. To store the string "avant-garde" in a C array of `char`, the array must have
 - (a) exactly 11 entries
 - (b) at least 11 entries
 - (c) at least 12 entries
 - (d) exactly 13 entries

3. Consider a program that contains the following statements:

```
int A[5] = { 1, 2, 3, 4, 5} ;
int B[5] = {101, 102, 103, 104, 105} ;

A = B ;
```

 - (a) This program would not compile.
 - (b) After execution of these statements, the array `A` would hold 101, 102, 103, 104 and 105.
 - (c) After execution of these statements, the array `B` would hold 1, 2, 3, 4 and 5.
 - (d) After execution of these statements, the variable `A` becomes a pointer to `int`.

4. A private data member of an object can be modified by code in
 - (a) the `main()` function.
 - (b) the function where the object was declared.
 - (c) member functions of the same class.
 - (d) all of the above

5. A public data member of an object can be modified by code in
 - (a) the `main()` function.
 - (b) member functions of another class.
 - (c) member functions of the same class.
 - (d) all of the above

6. A function with return type `void`
 - (a) does not return any values
 - (b) returns 0
 - (c) returns `'\0'`
 - (d) returns the `this` pointer

7. Global variables of type `int` are stored in
 - (a) the heap
 - (b) the data segment
 - (c) the stack
 - (d) the operating system

8. Local variables of type `int` are stored in
 - (a) the heap
 - (b) the data segment
 - (c) the stack
 - (d) the operating system

9. The `const` keyword can be used to
 - (a) define global const variables
 - (b) declare const member functions
 - (c) declare const parameters
 - (d) all of the above

10. A constructor for a class `Rover` is invoked when
 - (a) a local variable with type `Rover` is created.
 - (b) a global variable with type `Rover` is created.
 - (c) an array of `Rover` objects is defined.
 - (d) all of the above

II. Short Answers (4 points each)

1. Where in memory is a dynamically allocated `int` array stored?
2. Write down the function prototype of a function named `snap` that takes two `double` parameters and returns a `char`:
3. Write down the function prototype of a function named `crackle` that takes an array of `int` for its first parameter and a `char` for its second parameter and does not return any values.
4. What information is part of a function's signature?

5. Describe one use of the scope resolution operator `::`.

6. Describe one difference between a `const` member function and a non-`const` member function.

7. After the following code fragment, what are the values of `x`, `y` and `z`?

```
int x = 87, y = 92 ;
int &z = x ;

z = y ;
z = y - x ;
```

8. C header files usually begin with something like:

```
#ifndef _MONGOOSE_H
#define _MONGOOSE_H
```

Describe one purpose of these lines in the header file.

9. Consider the following function:

```
int bmw (int *ptr1, int *ptr2) {  
    int k = 7 ;  
  
    *ptr1 = k - *ptr2 ;  
    *ptr2 = k + *ptr1 ;  
    k = *ptr1 + *ptr2 ;  
    return k ;  
}
```

Suppose we call `bmw()` as follows:

```
int a = 12, b = 9, c = 4 ;  
  
c = bmw(&a, &b) ;
```

What are the values of `a`, `b` and `c` at the end of this program fragment?

10. Write down the contents of the array `A` after the following statements:

```
int A[5] = { 77, 92, 31, 46, 18} ;  
int *ptr ;  
  
ptr = A ;  
ptr = ptr + 2 ;  
*ptr = 999 ;
```

III. Coding (8 points each)

1. Assume that the `int` variable `n` contains a positive value.

Write a syntactically correct C++ program fragment, which must include a `for` loop, that prints out the first `n` odd numbers followed by the sum of the first `n` odd numbers. For example, if `n` has the value 3, your program should print out:

$$1 + 3 + 5 = 9$$

If `n` is 9, then your program should print out:

$$1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 + 17 = 81$$

You do have to pay attention to C++ syntax. Declare all variables.

2. Write a syntactically correct `while` loop in C/C++ that repeatedly asks the user for a number as input until the user enters a number that is divisible by 3. You do not have to worry about the user entering non-numerical or badly formatted input.

You do have to pay attention to C++ syntax. Declare all variables.

3. Write a syntactically correct C++ function called `space2plus` that takes a `char` array as a parameter and returns an `int` value. Your function should expect the `char` array to be a null-terminated C string. The function `space2plus` must replace every space character found in the parameter and with a plus symbol. The function should return the number of replacements performed.

For example, after the sequence of statements:

```
int n ;
char str[] = "The referee's gun starts to roar" ;
n = space2plus(str) ;
```

the variable `n` should be 5 and `str` should hold the string `"The+referee's+gun+starts+to+roar"`.

You do have to pay attention to C++ syntax. You must not use any global variables.

4. Write a syntactically correct C++ function called `sort` that takes two parameters. After calling `sort` with two `int` variables, the first variable should hold the larger of the two arguments, and the second variable should have the smaller of the two arguments. The function `sort` should not return any values.

For example, after the sequence of statements:

```
int a = 5, b = 9 ;  
sort(a, b) ;
```

the variable `a` should be 9 and the variable `b` should be 5. On the other hand, after:

```
int c = 15, d = 9 ;  
sort(c, d) ;
```

the variable `c` is still 15 and the variable `d` is still 9.

You do have to pay attention to C++ syntax. You must not use any global variables.