Definition Matching (8 points)

1. (8 pts) Match the words with their definitions. Choose the best definition for each word.

   Lifetime ______
   Scope ______
   Hierarchical Structure ______
   Value Parameter ______
   Static Variable ______
   Member Selector ______
   Aggregate Operation ______
   Function Prototype ______

A) Definition is not listed below (can be used more than once if necessary)

B) A variable for which memory remains allocated throughout the execution of the entire program.
C) The DataType of the result value returned by a function.
D) A parameter that receives the location (memory address) of the corresponding argument

E) A structured collection of components, all of the same DataType, that is given a single name.
F) A function declaration that includes the body of the function
G) A parameter that receives a copy of the value of the corresponding argument.

H) The expression used to access components of a struct variable.
I) The region of a program where it is legal to reference (use) an identifier.
J) The period of time during program execution when an identifier has memory allocated to it

K) A structure in which at least one of the members is itself a structure
L) A Function declaration without the body of the function
M) An operation on a data structure as a whole, as opposed to an operation on an individual component of the data structure.
True or False – (10 Points)

2. (10 pts) Circle T for true and F for false:

T   F   a) A break statement causes the program to proceed to the end of the current loop?

T   F   b) Static variables retain their value from function call to function call.

T   F   c) When a continue statement is executed, the innermost loop in which it appears is exited.

T   F   d) Arguments corresponding to value parameters can be literal values.

T   F   e) Local identifiers have name precedence over global identifiers.

T   F   f) void functions must use return expression;

T   F   g) Members of a structure must have unique names

T   F   h) Members of a structure can be of different DataTypes.

T   F   i) The expression date.day is used to access the day member of the date structure.

T   F   j) The lifetime of a local variable is for the duration of the program

Multiple choice (7 points) – Questions 3 – 9

For these problems circle all correct answers.

For example if answers A, C and E are all valid then circle A, C and E.

3. How many function values does a value returning function have?

A) 0   B) 2   C) As many as necessary

D) 1   E) 3   F) None of these

4. Which operations below ARE ALLOWABLE aggregate operations on structures?

A) Input/Output   B) Assignment   C) Arithmetic

D) Return as a functions return value   E) Comparison
5. **Reference parameters** (passing by reference) are used if a parameters data flow is

A) One-way, into the function  
B) One-way, out of the function  
C) Two-way, into and out of the function  
D) B and C  
E) None of these

6. Which parameters in the following **function heading** are **value** parameters?

```c
void DoSomething(string date, int& num, float& average, float sum, string name)
```

A) date  
B) num  
C) average  
D) sum  
E) name  
F) None of them

7. Which parameters in the following **function heading** are **reference** parameters?

```c
void DoSomething(string  date, int& num, float& average, float sum, string name)
```

A) date  
B) num  
C) average  
D) sum  
E) name  
F) None of them

8. (2 pts) A function that **returns a function value** is known as what kind of function?

A) Value returning  
B) Reference Parameter  
C) Empty  
D) Void  
E) Expression less  
F) None of these

9. The **void** function named **GetNums** has two parameters

A pass-by-value parameter named x of type float  
A pass-by-value parameter named num of type int.

Which of the following choices is the most appropriate **function heading** for the function **GetNums**? (There is only one answer on this problem)

A) void GetNums( float&, int )  
B) void GetNums( float x, int num )  
C) void GetNums( float x, int num )  
D) void GetNums( float& x, int& num )  
E) none of the above
10. (6 pts) Given the following constant and variable definitions/declarations.

```c
const int SUM = 10;
int size;
float average;
string name;
int square(float); // function prototype
```

and the following list of expressions to be used as arguments in a function call:

a) name  
b) ‘A’  
c) size  
d) SUM*10  
e) square(average)  
f) average  
g) “name”  
h) average*float(sum)

A) Which expressions above are valid for use as arguments with value parameters?

B) Which expressions above are valid for use as arguments with reference parameters?

11. (6 pts) There are two functions shown in the code segment below. Assume all variables and function prototypes have been correctly declared before this segment of code.

```c
Average(sum, num, average);
Name = ReadInfo(inFile, myStruct);
```

A) Which function(s) is(are) most likely value-returning function(s)?

B) Which function(s) is(are) most likely void function(s)?

C) What are the arguments that are used in the function calls.
12. (8 pts) When the program shown below is executed, what is the output to the screen? This problem deals with the scope of a variable in a program, and the order of execution of statements.

```
#include <iostream>
using namespace std;

void function_A(int);
void function_B(int&);
int number = 1; // global variable declaration of number

int main()
{
    int number = 2;
    cout << "number in main is: " << number << endl;
    function_B(number);
    return 0;
}

void function_A(int number)
{
    number = number + 2;
    cout << "number in function A is: " << number << endl;
}

void function_B(int& sum)
{
    sum = sum - 1;
    cout << "number in function B is: " << number << endl;
    function_A(number);
    cout << "sum in function B is: " << sum << endl;
}
```

The identifying phrases written by the cout statements in this program are shown below. In the blank to the left of the lines, place 1, 2, 3 or 4 to indicate the order the statements are printed (1 for first, 4 for last). The value output is placed in the blank at the end of the line.

___ number in main is: ___
___ number in function A is: ___
___ number in function B is: ___
___ sum in function B is: ___
13. (5 pts) Consider the following structure declarations when answering the questions below.

```c
struct Date
{
    int day;
    int month;
    int year;
};
struct Account
{
    string name;
    Date dueDate;
};
```

a) Write a statement that declares the identifier `today` as a variable of DataType `Date`.

b) Write a `cout` statement that will output the value of `year` of the variable `today`.

c) Write a statement that declares the identifier `kohls` as a variable of DataType `Account`.

d) Write a statement that assigns a value of “John” to the `name` member of `kohls`.

e) Write a `cout` statement that will output the value of `day` of the `dueDate` member of `kohls`.

14. (4 pts) Write the type declaration for a struct DataType named `LogType` containing the following members:

- an integer variable representing the log entry number
- a string variable representing the name of the person making the log entry
- a floating-point variable indicating the cost of the entry
- A Date variable indicating the day of the entry where `Date` is a structure already defined
15. (10 pts) Finish the program below by adding a void function as specified below. **Add only a function prototype, function call statement and function definition** to the following program.

- The name of the **void** function is **InitStruct**.
- The function has one parameter of the struct **DataType Date**.
- The function initializes the members of the parameter with values corresponding to the date January 1, 2000 (use 1 to represent January)
- The information stored in the parameter **must be available in main()** after the function call.

```cpp
#include <iostream>
using namespace std;

struct Date
{
    int month;
    int day;
    int year;
};

// Place the function prototype below this line

int main()
{
    Date date;

    // Place the function call statement below this line

    return 0;
}

// Place the function definition below this line
```
16. (8 pts) Rewrite the **value returning function definition** below as a **void function definition** such that the caller of the function still has access to the result that is being returned by the value returning function. **Be sure to change the function return type from string to void and to add in another parameter.**

```cpp
string FindSubstring()
{
    string substring, line;
    cout << "Enter a line: ";
    getline(cin, line);
    substring = line.substr(line.size()/2, line.size());
    return substring;
}
```

17. (8 pts) Rewrite the **void function definition** below as a **value returning function definition**. The value returned by the void function using the reference parameter is to become the return value of the value returning function.

- **Do not use any reference parameters with the value returning function.**
- **Two parameters only are allowed for use with the value returning function.**
- **You can add a single variable declaration if necessary.**

```cpp
void Calculate (float& result, float num1, float num2)
{
    result = (num1 + num2)*5;
}
```
18. (12 pts) Write a Boolean value-returning function definition called OpenFile with 2 parameters.

- The function prompts the user for the name of the file to open and opens the file (use .c_str()).
- The function returns a value of true if the file was successfully open.
- A value of false is returned if the file was not successfully opened.

- The first parameter contains the name of the file that is entered for opening. The name of the file entered must be available for use by the caller of the OpenFile function.
- The second parameter, an ifstream variable, represents the input stream to use.
19. (8 pts) For the following code segment, write out what is printed to the screen. Show the displayed output precisely by using the following rules:
- Write one character per box.
- Skip a box to indicate the presence of a blank space in the output.
- Skip a row to indicate the presence of a blank line in the output.

```cpp
#include <iostream>
using namespace std;

void Test();

int main()
{
    Test();
    Test();
    Test();
    Test();
    return 0;
}

void Test()
{
    int i = 0;
    static int j = 5;

    i++;
    cout << i << "—" << j << endl;
    j++;
}```
Extra Credit (5 pts) The following program is executed. The user enters the integer “7” when prompted for a number. What is the output to the screen?

Be careful on this problem. Think about what is being performed with which variables.

```cpp
#include <iostream>
using namespace std;
void summation( int& num, int result );
int main ()
{
    int number = -1;
    int result = -1;
    cout << "Enter an integer between 1 and 10: ";
    cin >> number;
    summation(number, result);
    cout << "Summation for " << number << " is: " << result << endl;
    return 0;
}
void summation( int& num, int result )
{
    int loop;
    int result = 0;
    for (loop = num; num >=1; num--)
    {
        result = result + loop;
        num = num -1;
        cout << "result: " << result << " - " << num << endl;
    }
    return;
}
```