**Definition Matching – (10 Points)**

1. (10 pts) Match the words with their definitions. Choose the **best definition** for each word.

   - Function Prototype _____
   - Scope _____
   - Event Counter _____
   - Local Variable _____
   - Argument _____
   - Name Precedence _____
   - Switch Expression _____
   - Reference Parameter _____
   - Function Call _____
   - Count-Controlled Loop ___

   A) Definition is not listed below (may be used more than once)

   B) A statement that transfers control to a function.
   C) Computes a new value by performing a specified set of operations on given values.
   D) A variable declared within a block and not accessible outside of that block
   E) The region of program code where it is legal to reference (use) an identifier.

   F) A function declaration without the body of the function
   G) A variable or expression listed in a call to a function
   H) A parameter that receives a copy of the value of the corresponding argument
   I) A counter variable that is incremented with each iteration of the loop

   J) The order in which the computer executes the statements in a program
   K) A control structure that causes a statement or group of statements to be executed repeatedly.
   L) The period of time during program execution when an identifier has memory allocated to it.
   M) The expression whose value determines which switch label is selected.

   N) The precedence that a local identifier in a function has over a global identifier with the same name
   O) A loop that terminates when something inside the loop body signals that the loop should be exited.

**True or False – (10 Points)**

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

   T  F   a) A **while loop** is executed one or more times.

   T  F   b) When a function exits, the contents of any **static variables** are destroyed (lost).

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

   T  F   d) Arguments corresponding to **value parameters** can be literal values.
The body of a *for* loop executes zero or more times.

Count-Controlled loops are executed a pre-specified number of times.

Reference parameters receive a copy of an arguments value

A relational expression consists of logical operators and Boolean variables only.

A function call can contain more arguments than the number of parameters in the corresponding function heading.

Value returning functions cannot use *return* expression;

---

**Multiple choice (10 points) – Questions 3 – 7**

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. Which of the following loop types are examples of an Event-Controlled loop?

A) Sentinel-Controlled       B) End-of-File-Controlled       C) Count-Controlled

D) Flag-controlled       E) None of these

4. Assume that the Boolean variable *X* has the value *false* and that the Boolean variable *Y* has the value *false*. What is the value of the following logical expression?

\[(Y \lor (X \lor \neg X))\]

A) true       B) false       C) “true”       D) “false”       E) None of the above

5. Assume that the Boolean variable *X* has the value *true* and that the Boolean variable *Y* has the value *false*. What is the Boolean value of the following logical expression?

\[(\neg (X \lor \neg Y) \&\& (X \&\& (Y \lor \neg Y))) \lor X\]

A) true       B) false       C) “true”       D) “false”       E) None of These
6. The \texttt{void} function named \texttt{GetNums} has two parameters

   A \texttt{pass-by-reference} parameter named \texttt{x} of type \texttt{float}
   A \texttt{pass-by-value} parameter named \texttt{num} of type \texttt{int}.

Which of the following choices is a valid \texttt{function heading} for the function \texttt{GetNums}?

   A) \texttt{void GetNums( float& , int )}
   B) \texttt{void GetNums( float& x , int num )}
   C) \texttt{void GetNums( float x , int num )}
   D) \texttt{both A and C}
   E) \texttt{none of the above}

7. The \texttt{void} function named \texttt{GetNums} has two parameters

   A \texttt{pass-by-reference} parameter named \texttt{x} of type \texttt{float}
   A \texttt{pass-by-value} parameter named \texttt{num} of type \texttt{int}.

Which of the following choices is a valid \texttt{function prototype} for the function \texttt{GetNums}?

   A) \texttt{void GetNums( float , int );}
   B) \texttt{void GetNums( float x , int num );}
   C) \texttt{void GetNums( float& x , int num );}
   D) \texttt{both A and C}
   E) \texttt{none of the above}

---

\textbf{Short Answer (120 points) – Questions 8 – 23}

8. \textbf{(8 pts)} Given the following constant and variable definitions/declarations.

\begin{verbatim}
const int MAX = 100;
int sum;
float average;
string name;
int square(int); // function prototype
\end{verbatim}

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

\begin{itemize}
  \item a) average
  \item b) ‘A’
  \item c) sum
  \item d) square(sum)
  \item e) MAX
  \item f) name
  \item g) “name”
  \item h) sum - average
\end{itemize}

\textbf{A)} Which expressions are valid for use as arguments with value parameters?

\textbf{B)} Which expressions are valid for use as arguments with reference parameters?
9. (8 pts) There are three functions shown in the code segment below. Assume all variables and function prototypes have been correctly declared before this segment of code.

\[
\text{Average}(\text{Sum}(x), \ \text{avg}); \\
A = \text{Square}(w, \ y);
\]

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 five different arguments that are used in the function calls.

10. (8 pts) If the numbers entered are 1 3 5 7 9, what is the output for the following segment of code? Assume all variables are declared as integers.

\[
\text{sum} = 0; \\
\text{cin} \ >> \text{number}; \\
\text{do} \\
\{ \\
\quad \text{sum} = \text{sum} + \text{number}; \\
\quad \text{cin} \ >> \text{number}; \\
\}\text{while} (\text{number} < 9); \\
\text{cout} \ll \text{sum};
\]
For Problems 11 and 12 show precisely the displayed output

- 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.

11. (8 pts) What is the output for the following segment of code? All variables are integers

```cpp
int count = 0;
while ( count < 20 )
{
    count++;>
    if (count%4 != 0)
        continue;
    cout << "-" << count << "-" << endl;
}
```

12. (8 pts) What is the output for the following segment of code? All variables are integers

```cpp
for (i = 0 ; i < 10; i++)
{
    for (j = 0; j < 10; j++)
    {
        cout << i << j;
        if ( j >= 2)
            break;
    }
    cout << endl;
    if ( i >= 1)
        break;
}
```
13. (6 pts) What is the output of the following program?

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

void Sum(int&, int);

int main()
{
    int sum=-1, num=5;
    Sum(num, sum);
    cout << "Sum of integers from 1 to " << num << " is: " << sum << endl;
    return 0;
}

void Sum(int& n, int sum)
{
    while(n>=1)
    {
        sum=sum+n;
        n=n-1;
    }
}
```

Sum of integers from 1 to ____________ is: ______________

14. (8 pts) Write a do-while loop that continues to prompt for and read an integer value entered by the user until the user enters in a positive integer. In other words the do-while loop continues to execute until the user enters in a positive integer (number greater than 0).
15. (10 pts) Finish the code segment below so that the following actions are performed:
   a) Prompt the user for the name of an input file and read it.
   b) Open the file.
   c) If the file did not open successfully, print out a message stating as such, reset the input
      stream variable (code to do this is inFile.close(); inFile.clear();)
   d) Repeat steps a,b,c until a file is successfully opened or the user enters ctrl-c.

```cpp
ifstream inFile;
string filename;
cout << “Enter the name of the input file (or ctrl-c to exit): “;
cin >> filename;
inFile.open(filename.c_str());
// rest of the code segment goes below this line
```

16. (6 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 (contained in the function
    variable sum) that is being returned by the value returning function

    - Use three parameters (one reference and two value) with the void function.

```cpp
float FindSum(float average, int num_values)
{
    float sum;
    sum = average*num_values;
    return sum;
}
```
17. (4 pts) What is the output of the following code segment if `num` has a value of 50?

```cpp
if (num > 20)
    if (num < 40)
        cout << "num is less than 40";
    else
        cout << "num is less than 20";
else
    cout << "end" << endl;
```

18. (4 pts) What is the output of the following code segment?

```cpp
int a = 10;
if (a = 0)
    cout << "a is zero";
else
    cout << "a is not zero";
```

19. (8 pts) Finish the segment of code using an if-then-else-if structure such that the following phrases are printed out for the speeds indicated:

- “Too Slow” if speed is less than 30
- “Just Right” if speed is less than 60 and greater than or equal to 30
- “Too Fast” if speed is greater than or equal to 60

```cpp
int speed;
cout << "Enter in a speed (integers only): ";
cin >> speed;
// Place if-then-else-if testing structure below this statement
```
20. (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.

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

void function_A(int);
void function_B(int);
int number = 1;

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

void function_A(int num)
{
    int number = 3;
    num = num + 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(sum);
    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). In The blank at the end of the line, put in the output value.

___ number in main is: ___
___ number in function A is: ___
___ number in function B is: ___
___ sum in function B is: ___

21. (8 pts) Consider the following segment of code
int value;
cin >> value;
switch (value)
{
    case 1: cout << "One";  
    case 2: cout << "two";  
        break;
    case 3: cout << "Three"; 
    case 4: cout << "Four";  
        break;
    default: cout << "No Match"; 
}
a) What is the output if the value entered is 10?

b) What is the output if the value entered is 3?

22. (8 pts) Consider the following segment of code

int value;
cin >> value;
switch (value)
{
    case 7: cout << "7";  
    case 8: cout << "8";  
    case 9: cout << "9";  
    case 10: cout << "10";  
    default: cout << "No Match"; 
}

a) What is the output if the value entered is 1?

b) What is the output if the value entered is 7?
23. (10 pts) Add a function prototype, function call statement and function definition only to the following program. No other information is to be added (i.e. variables).

- The name of the Boolean value-returning function is **OpenInput**.
- The function has one parameter – an input file stream.
- The function is to prompt the user for the name of the input file, open that input file and return the status of the input file stream via the functions return value.

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

// Place the function prototype for OpenInput below this line

int main()
{
    ifstream inputFile;
    bool status;

    // Place the function call statement for OpenInput below this line
    // store the return value in the variable status

    cout << "Status: " << status << endl;
    return 0;
}
// Place the function definition for OpenInput below this line
```