Bonus Question (+4 pts):  
The following bonus question is for students present in class the day the code was given.

Possible answers for the 4 questions are:

A) 123ABC  B) EFG456  C) XYZ123  D) 456ABC

Bonus #1 (+2 pts) Code #1 for Test2 is _______________________

Definition Matching – (12 Points)

1. (12 pts) Match the words with their definitions. Choose the best definition for each word.

Function Prototype _____ Loop _____ Event Counter _____
Local Variable _____ Iteration Counter _____ Name Precedence _____
Switch Expression _____ Reference Parameter _____ Function Call _____
Argument _____ Scope _____ Event-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 parameter that receives a copy of the value of the corresponding argument
E) A counter variable that is incremented with each iteration of the loop

F). A Function declaration without the body of the function
G) The order in which the computer executes the statements in a program
H) A control structure that causes a statement or group of statements to be executed repeatedly.
I) A variable or expression listed in a call to a function

J) A variable declared within a block and not accessible outside of that block
K) The region of program code where it is legal to reference (use) an identifier.
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 – (15 Points)

2. (15 pts) Circle T for true and F for false:

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

T   F  b) A compile error occurs if the DataType defining a function type is omitted.

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

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

T   F  e) The body of a for loop executes at least once

T   F  f) Event-Controlled loops are executed a pre-specified number of times.

T   F  g) Value parameters receive a copy of an arguments value

T   F  h) A logical expression consists of logical operators and Boolean variables only.

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

T   F  j) A switch statement MUST have a default switch label.

T   F  k) void functions cannot use return expression;

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

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

T   F  n) While loops can be nested (one loop inside of another loop)?

T   F  o) Local variables can be accessed outside of the block in which they are declared?
Multiple choice (20 points) – Questions 3 – 12
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 is not an example 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 true and that the Boolean variable \( Y \) has the value false. What is the value of the following logical expression?

\[(Y \text{ || } (X \text{ || !X}))\]

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

5. What is the output for the following segment of code, if the numbers entered are 1 3 5 7 9? Assume all variables are declared as integers.

\[
\begin{align*}
\text{sum} & = 0; \\
\text{cin} & \ >> \ \text{number}; \\
\text{do} & \ \\ & \{ \\
& \quad \text{sum} = \text{sum} + \text{number}; \\
& \quad \text{cin} > \rightarrow \text{number}; \\
& \} \text{while} (\text{number} \leq 5); \\
\text{cout} & \ll \ \text{sum};
\end{align*}
\]

A) 8 B) 9 C) 15 D) 16 E) None of These

6. The void function named GetNums has two parameters

a pass-by-value parameter named \( x \) of type float
a pass-by-reference parameter named \( \text{num} \) of type int.

Which of the following choices is a valid function heading for the function GetNums?

A) void GetNums( float , int& )
B) void GetNums( float \( x \) , int num )
C) void GetNums( float \( x \) , int& num )
D) both A and C
E) none of the above
7. The **void** function named **GetNums** has two parameters

- a **pass-by-value** parameter named \( x \) of type **float**
- a **pass-by-reference** parameter named **num** of type **int**.

Which of the following choices is a valid **function prototype** for the function **GetNums**?

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

8. What is the output of the following code segment if num has a value of 2? Assume all variables are integers.

```cpp
switch(num)  
{  
    case 1: cout << "a";  
    case 2: cout << "b";  
        break;  
    case 3: cout << "c";  
    default: cout "end";  
}
```

A) bcend  
B) bc  
C) a  
D) b  
E) None of These

9. What is the output of the following code segment if value has a value of 2? Assume all variables are integers.

```cpp
switch(num)  
{  
    case 1: cout << "a";  
    case 2: cout << "b";  
    case 3: cout << "c";  
    default: cout "end";  
}
```

A) bcend  
B) c  
C) abcend  
D) end  
E) None of These
10. Assume that the Boolean variable \( X \) has the value \textit{false} and that the Boolean variable \( Y \) has the value \textit{true}. What is the Boolean value of the following logical expression?

\[
( !(X \lor !Y) \land (X \land (Y \lor !Y)) ) \lor X
\]

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

11. What is the output value of score for the following code segment?

```c
int score = 95;
if (score > 90)
    score = score - 5;
else if (score > 80)
    score = score - 5;
else if (score > 70)
    score = score - 5;
score = score - 5;
cout << score << endl;
```

A) 70  B) 75  C) 80  D) 85  E) None of These

12. What is the output for the following segment of code? Assume all variables are declared as integers

```c
sum = 0;
for (j = 1 ; j <= 6; j++)
{
    if ( j == 2 || j == 4 )
        continue;
    sum = sum +j;
}
cout << sum;
```

A) 0  B) 1  C) 3  D) 6  E) 9  F) 15  G) 18  H) 21  I) None of These
13. (2 pts) Provide the names of two types of **event-controlled** loops

14. (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: ______________

15. (6 pts) Given the following constant and variable definitions/declarations.

```cpp
const int MAX = 100;
int sum;
float average;
string name;
int square(int); // function prototype
```

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

a) sum   b) sum-25   c) “Hello”   d) 22.5 

e) square(sum)   f) name   g) sum   h) MAX 

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

B) Which expressions are valid for use as arguments with reference parameters?
16. (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
#include <iostream>
#include <fstream>

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

17. (4 pts) What is the output of the following code segment if \texttt{num} has a value of 50?

```cpp
if (num > 20)
    if (num < 40)
        cout << “num is less than 40-“;
    else
        cout << “num is unknown-“;
else
    cout << “end\n“;
```

18. (8 pts) For the following code segment, 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.

```cpp
int loop_b;
int loop_a = 0;

while ( loop_a <= 3 )
{
    for (loop_b = loop_a; loop_b <= 4; loop_b++)
        cout << loop_b;
    cout << loop_a << endl;
    loop_a++;
}
```
19. (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;
int main()
{
    int number = 2;
    cout << "number in main is: " << number << endl;
    function_B(number);
    return 0;
}
void function_A(int& num)
{
    int number = 3;
    cout << "number in function A is: " << number << endl;
    num = num - 1;
}
void function_B(int sum)
{
    cout << "number in function B is: " << number << endl;
    function_A(sum);
    cout << "sum in function B is: " << sum << endl;
    sum = sum - 1;
}
```

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

Page 9 of 13
20. (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 average) that is being returned by the value returning function.

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

```c++
float FindAverage(float sum, int num_values)
{
    float average;
    average = sum/num_values;
    return average;
}
```

21. (8 pts) Finish the segment of code using an if-then-else-if structure such that a letter grade is printed out for the grade entered. Letter grade assignments are based on the following information: A for grade > 90, B for grade >76, but <= 90, C for grade > 60, but <=76 and F for grade <=60.

```c++
int grade;
cout << "Enter a students grade (integers only): ";
cin >> grade;
// Place if-then-else-if testing structure below this statement
```
22. (10 pts) Consider the following segment of code

```cpp
int count = 10;
int sum = 1;
while (count > 0)
{
    sum += sum;
    cout << sum << endl;
    count--;
}
```

a) Rewrite the above code segment as a for loop such that the same output is obtained.

```cpp
for (int i = 10; i > 0; i--)
{
    int sum = 1;
    cout << sum << endl;
    sum += sum;
}
```

b) Rewrite the above code segment as a do-while loop such that the same output is obtained.

```cpp
int sum = 1;
do
{
    cout << sum << endl;
    sum += sum;
} while (count > 0);
```
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 value-returning function is Average.
- The function has two parameters of data type float.
- The function is to divide the first parameter by the second parameter and return the result.

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

// Place the function prototype for Average below this line

int main()
{
    float avg;
    float sum = 225;
    float num = 10;

    // Place the function call statement for Average below this line
    // use sum as the first argument and num as the second argument
    // store the return value in the variable avg

    cout << "Average: " << avg << endl;
    return 0;
}
// Place the function definition for Average below this line
```
EXTRA CREDIT) (8 pts) Consider the following function definition:

```c
int product(int& num1, int& num2, int num3)
{
    return num1*num2*num3;
}
```

Write a complete program that (you do not have to rewrite the above function definition):

1) in `main()` prompts the user for three integers and places them in variables,
2) in `main()` calls the `product` function using the three integers entered as arguments, and
3) in `main()` prints the three integers and the return value of the function.
4) Assume that the function definition is to appear below `main`.

NOTE: This is an entire program, so show all `#include` directives, function prototypes, variable declarations, etc. You do not have to rewrite the function definition.