Exam 3 Instructions – Time Limit 60 Minutes

**Turn in all exam papers and the bubble sheet**

**General Instructions:**
- Neatness counts!! If the machine cannot read your answer, you will receive no credit.
- This is a closed book/closed notes exam.
- No reference materials of any kind will be permitted.
- No calculators or other portable computing/data storage devices are permitted.
- Any C++ code segments that you write must be syntactically correct or you will lose points.

There are two parts to this exam:

- **Part I** (45 pts) - Bubble Sheet – True/False, Multiple Choice, Vocabulary, fill in the blank
- **Part II** (55 pts) - Written part – Short answers, writing code segments, algorithms and Functional Decompositions

**Bubble Sheet Instructions:** (Part I)

1) Use a #2 pencil to complete the bubble sheet!

2) Print your name in the Name box as follows:

   LASTNAME  FIRSTNAME  MIDDLEINITIAL

   Example:  Bowman  Ron  D

3) Use your #2 pencil to fill in the corresponding bubbles under each character of your name. (Bubble marking instructions are included on Side 2 of the bubble sheet.)

4) In the identification number area, write in (starting at the left) the numerical part of your “A” number and bubble in the corresponding bubbles under each number.

5) Record your answers to the questions in Part I on Side 1 of the bubble sheet. Neatness Counts!!

**Warnings:**
- For Part I, answers not recorded on the bubble sheet will receive no credit!!
- Be wary of skipping problems!!
- I recommend that you answer each question in Part I in the order presented.
- Read all possible answers. Some questions have answers like “a and b” or “all of the above”

**Instructions for Part II – Short answers**

1) Write your answers as neatly as possible in the space provided.
2) In some cases it is best to think for a minute and then start writing.
Exam 3 – Part I (45 pts): Multiple Choice
(Select the best answer for each problem and record it on the bubble sheet)
(You may bubble in more than one letter for each question. If answers A and C are correct, then bubble both of them in. Also, some questions may have answers with choices of AB, BC, etc., so bubble in both letters)

Read the above directions

TRUE/FALSE Section: Bubble in A for True and B for False

1. True or False? It is not an error if the DataType defining a function type is omitted.
2. True or False? Reference parameters receive a copy of an arguments value.
3. True or False? An argument corresponding to a reference parameter can be a constant or arbitrary expression?
4. True or False? The use of the statement: return; is valid in a void function
5. True or False? A function call can contain more arguments than the number of parameters in the corresponding function heading.
6. True or False? Static variables retain their value from function call to function call.
7. True or False? Local identifiers have name precedence over global identifiers
8. True or False? Do-While loops cannot be nested (one loop inside of another loop)?
9. True or False? A do-while loop is executed 0 or more times
10. True or False? When a continue statement is executed, the innermost loop in which it appears is exited.
11. True or False? All possible values for the switch expression must be included among the case labels for a given switch statement.
12. True or False? A switch statement MUST have a default switch label
13. True or False? Input/Output operations are allowable aggregate operations on structures?
14. True or False? Members of a structure do not have to be of the same DataType.
15. True or False? A variable declared as a union can hold more than one value at any given time?
Definition Matching: match the best definition for each word listed

16. Definition for Function Definition ____
17. Definition for Function value type ____
18. Definition for Function Prototype ____
19. Definition for Value Parameter ____
20. Definition for Argument ____

21. Definition for Name Precedence ____
22. Definition for Static Variable ____
23. Definition for Scope ____
24. Definition for Lifetime ____
25. Definition for Switch Expression ____

26. Definition for Aggregate Operation ____
27. Definition for Hierarchical Structure ____
28. Definition for Structure ____
29. Definition for Member Selector ____

A) Definition is not listed below (can be used more than once if necessary)
B) The precedence that a local identifier in a function has over a global identifier with the same name
C) The data type of the result value returned by a value returning function
D) The expression used to access components of a struct variable.
E) A parameter that receives the location (memory address) of the corresponding argument

For the choices below, bubble in ALL letters indicated

AB) A variable or expression listed in a call to a function
AC) A function declaration that includes the body of the function
AD) The region of program code where it is legal to reference (use) an identifier
AE) A variable for which memory remains allocated throughout the execution of the entire program.

BC) A variable declared within a block and not accessible outside of that block
BD) A parameter that receives a copy of the value of the corresponding argument.
BE) The period of time during program execution when an identifier has memory allocated to it

CD) A structured collection of components, all of the same DataType, that is given a single name.
CE) A Function declaration without the body of the function

DE) A structure in which at least one of the members is itself a structure

ABC) The expression whose value determines which switch label is selected.
ABD) An operation on a data structure as a whole, as opposed to an operation on an individual component of the data structure.
MULTIPLE CHOICE SECTION

For the multiple choice questions below, remember that multiple answers may be correct, and in those instances all correct answers need to be bubbled in on the answer sheet. For example, if answers A, C and E are all correct, then bubble in A, C and E.

30. A function that does not return a value is known as what kind of function?
   A) Value returning function  B) Reference Parameter  C) Empty
   D) Void function  E) Expression less  AB) None of these

31. How many function values does a value returning function have?
   A) 1  B) 2  C) 3  D) 4  E) As many as necessary  AB) None of these

32. 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

33. Value parameters (passing by value) 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) None of these

34. 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) None of these

35. Which of the following can be used as a switch expression? (Select all correct answers):
   A) bool variable  B) string constant  C) char variable  
   D) integer variable  E) floating point variable  AB) None of Them
36. When the following code is executed, how many iterations of the loop are performed?

```c
int number = 2;
bool notDone = true;
do
{
    number=number*2;
    notDone = (number > 15);
}while (notDone)
```

A) 0  B) 1  C) 2  D) 3  E) 4  AB) Nothing, it is an infinite loop  AC) None of These

37. Which parameters in the following function heading are value parameters?

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

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

38. Which parameters in the following function heading are reference parameters?

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

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

39. A function SomeFunc has two parameters, alpha and beta, of type int. The data flow for alpha is one-way, into the function. The data flow for beta is two-way, into and out of the function. What is the most appropriate function prototype for SomeFunc?

A) void SomeFunc( int alpha, int beta );
B) void SomeFunc( int& alpha, int beta );
C) void SomeFunc( int& alpha, int& beta );
D) void SomeFunc( int alpha, int& beta );
E) None of the above

40. What is the output of the following code segment if num has a value of 1? Assume all variables are integers.

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

A) abcend  B) abc  C) ab  D) a  E) b  AB) c  AC) cend  AD) None of these
Problems 41 – 43: a C++ void function named SumOfSquares will sum the squares of the integers from 1 through N inclusive and return the integer sum via a parameter. (Preconditions: N assigned a valid integer value and N >= 1) N will be an arbitrary value of type int supplied by the calling program via an argument.

41. Based on the description above, which of the following choices is the most appropriate function prototype for the function SumOfSquares?

A) void SumOfSquares(int, int);
B) void SumOfSquares(int&, int);
C) int SumOfSquares(int&, int);
D) int SumOfSquares(int, int);
E) None of the above

42. Based on the description above, which of the following choices is the most appropriate function heading for the function SumOfSquares?

A) void SumOfSquares(int, int)
B) void SumOfSquares(int&, int)
C) int SumOfSquares(int&)
D) int SumOfSquares(int)
E) None of the above

43. Based on the description above, which of the following choices is the most appropriate function call for the function SumOfSquares? (Assume all variables are integers and N has a value >= 1)

A) SumOfSquares(N,25);
B) SumOfSquares(sum, N);
C) sum = SumOfSquares( sum, 25);
D) sum = SumOfSquares(N,25);
E) None of the above

44. The void function named GetNums has two parameters:

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

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

A) void GetNums( float , int& );
B) void GetNums(float&, int);
C) void GetNums(float&, int&);
D) void GetNums( float x , int num );
E) void GetNums( float x , int& num );
AD) None of the above
45. The **void** function named **GetNums** has two parameters:

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

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

D)   void GetNums( float x , int num )
E)   void GetNums( float x , int& num )
AB) void GetNums(float& x, int num)
AC) void GetNums(float& x, int& num)
AD) None of the above
Exam 3 – Part II (55 pts): Short Answers
(Write your answers in the space provided)

For Bonus questions #1, #2 and #3 – circle the correct answer
Bonus #1 (+2 pts) The first code given out in class (for those who attended) is
A) SPR01  B) CPE112-01  C) CPE112-03  D) SPR06
E) CPE01  F) CPE02  G) XYZ-789  H) ABC123

Bonus #2 (+2 pts) The second code given out in class (for those who attended) is
A) SPR01  B) CPE112-01  C) CPE112-03  D) SPR06
E) CPE01  F) CPE02  G) XYZ-789  H) ABC123

Bonus #3 (+2 pts) The third code given out in class (for those who attended) is
A) SPR01  B) CPE112-01  C) CPE112-03  D) SPR06
E) CPE01  F) CPE02  G) XYZ-789  H) ABC123

1) (6 pts) Consider the code segment shown when answering the questions below:

```cpp
int number;
cout << "Enter an integer between 0 and 10: ";
cin >> number;
switch(number*2)
{
    case 2:  cout << 'A';
    case 3:  cout << 'B';
    case 5:  cout << 'C';
    case 8:  cout << 'X';
    case 13: cout << 'Y';
    case 16: cout << 'Z';
        break;
    default: cout << "default" << endl;
}
```

a) What is the output if 3 is entered?

b) What is the output if 8 is entered?
2) (6 pts) What is output (fill in the blanks) by the `cout` statement in `main` in the following program?

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

void SumInts(int& , int );

int main()
{
    int number, sum = 0;
    number = 5;
    SumInts(number, sum);
    cout << "The sum of " << number << " is: " << sum << endl;
}

void SumInts(int& num, int sum)
{
    sum = 0;
    do
    {
        sum = sum + num;
        num = num - 1;
    } while ( num >= 1)
}

The sum of ________ is: ________
```

3) (4 pts) Given the `for` loop below, write an equivalent `do-while` loop.

```cpp
for(int count=0; count <= 10;count++)
    cout<< " count is: " << count << endl;
```
4) (10 pts) Consider the following structure declarations when answering the questions below.

```c
struct Date
{
    int day;
    int month;
    int year;
};
struct Customer
{
    string item;
    string cust_name;
    Date date_purch;
};
```

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

b) Write a `cout` statement that will output the value of `month` of the variable `birthday`.

c) Write a statement that declares the identifier `John_Doe` as a variable of DataType `Customer`.

d) Write a statement that assigns a value of “Coat” to the `item` of `John_Doe`.

e) Write a `cout` statement that will output the value of `year` of the `date_purch` of `John_Doe`.

5) (5 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
6) (8 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. No other information is to be added (i.e. variables)

The name of the **void** function is **InitStruct**.
The function has one parameter of the struct **DataType Date**.
The function is to initialize each member of the structure with a value of 0.
The information stored in the structure 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
```
9) (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 = 2;
int main()
{
    int number = 4;
    function_B(number);
    cout << "number in main is: " << number << endl;
    return 0;
}
void function_A(int& num)
{
    int number = 1;
    num = num - 2;
    cout << "number in function A is: " << number << endl;
}
void function_B(int& sum)
{
    cout << "sum in function B is: " << sum << endl;
    function_A(sum);
    cout << "number in function B is: " << number << endl;
    sum = sum + 2;
}
```

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.

```plaintext
___ number in main is: ___
___ number in function A is: ___
___ sum in function B is: ___
___ number in function B is: ___
```
10) (4 pts) Rewrite the **void function definition** below as a **value returning function definition** such that the caller of the function still has access to the result that is being returned by the void function.

- Do not use any reference parameters with the value returning function.
- Use two value parameters with the value returning function only

```c
void Calculate (float& num1, float num2, float num3)
{
    num3 = num3*num3;
    num2 = num2*num2;
    num1 = num2 + num3;
}
```

11) (4 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 variable substring) that is being returned by the value returning function. (Hint: need to use a reference parameter for the void function)

```c
string FindSubstring(string line, string name)
{
    string substring;
    substring = line.substr(line.find(name),line.size());
    return substring;
}
```
Extra Credit) (2 pts) Consider the following code segment:

```cpp
int count = 0;
int number;
int sum = 0;
while (true)
{
    cout << "Enter any integer value: ";
    cin >> number;
    // The following expression is true if number is odd and positive
    if (number - (number/2)*2 == 1)
        break;
    else if (number > 0 )
        continue;
    else
        sum = sum + number;
    count++;
}
cout << count << " valid numbers were entered, and";
cout << " their sum is: " << sum << endl;
```

What is output to the screen by the last two `cout` statements (fill in the blanks shown below) if the numbers to be entered are: -1, 2, -5, -8, 6, 1, 0, -1, 3, -4, 1

(NOTE: the commas are not entered, they just show the separation of the numbers)

_______ valid numbers were entered, and their sum is: ________