cSE141 Introduction to Programming

Final Examination

Max Marks: 50 Time Allowed: 3 hours

Answer the questions in the spaces provided on the question sheets.
Please give clear and rigorous answers. Be to the point. Show your work.

Name:________ ERP: ____

Question:	True/False	MCQ's	Trace/Debug	Methods	Arrays	Recursion	Classes	Total
Marks:	10	9	11	5	5	5	5	50
Score:								

Question 1: True/False . . . . *10 marks*

True/False quesions. Circle the correct answer.

(a) **T F** A single std::cout statement in a C++ program could result inmultiple lines of output.

(b) **T F** Multiple `std::cout` statements in a C++ program could result ina single line of output.

(c) **T F** Variable names in C++ must have at least one upper-case letter.

(d) **T F** To end a C++ program both the `break` or the `exit` statement may be used.

(e) **T F** An if-else statement could be rewritten using two if statements without an else.

(f) **T F** The following statements give a run-time error in C++:

```
std::string name = "123";
 if (name.length() == 3) {
std::cout << name[2] << std::endl;
 }
```

(g) **T F** The output of the following statements is: `3 Done`

```
 int x = 2;
 int y = 1;
std::cout << x << y;
std::cout << " Done" << std::endl;
```

(h) **T F** To check and see if the value of variable `x` is between 3 and 9 we could use:

```
int x = 4;
if( 3 < x < 9) {
std::cout << "x is between 3 and 9" << std::endl;
}
```

(i) **T F** The output of the following lines of code is: `One Done`

```
int x = 2;
if( x = 1)
std::cout<<"One ";
std::cout<<""Done";
```

(j) **T F** The output of the statement below is: `1.7`

```
int x = 5;
int y = 3;
double answer = 0.0;
answer = x / y;
std::cout<< setprecision(1) << answer << std::endl;
```

Question 2: MCQ's .... *9 marks*

Circle the correct answer

(a) [1½ marks] Consider the program segment given below. (Remember that the % operator is the modulus operator, which gives the remainder after integer division.) Its output is:

```
   int i = 1;
  while (i <= 30) {
          i++;
         cout << i;
        if (i % 2 == 0) {
          cout << endl;}
            i++;
   }
```

i. All the numbers from 1 to 30, with a line break after the number 10

ii. Even numbers from 1 to 30 all one one line

iii. Odd numbers from 1 to 30, each on its own line

iv. Even numbers from 1 to 30, each on its own line

v. None of the above

(b) [2 marks] Consider method `second` shown below, which itself uses method `first`. For positive numbers, how would you best describe its return value?

```
int first(int x, int y) {
   int z=0;
   for (int i=0; i<y; i++) {
      z += x;
   }
   return z;
}

int second(int x, int y) {
   int z=1;
   for (int i=0; i<y; i++) {
      z = first( z, x);
   }
   return z;
}
```

i. $x + y$

ii. $x * x$

iii. $x * y$

iv. x^y

v. None of the above

(c) [1½ marks] The output of the following code in C++ is equal to:

```
int value = 2;
int limit = 5;
int result = 1;
for(int x=0; x<limit; x++) {
   result = result * value;
}
std::cout<<result;
```

i. $2 * 5$

ii. 2 + 5

iii. 25

iv. 32

v. None of the above

(d) [1 mark] If we start with the array `6 1 14 10 5 12 11 9`, which sorting algorithms will encounter `1 5 6 10 14 12 11 9` during the sorting process?

i. Insertion sort

ii. Mergesort

iii. All of the above

iv. None of the above

(e) [1 mark] In worst-case, the number of steps taken by *binary search* when applied on an array of size N is proportional to

i. constant

ii. $\log N$

iii. N

iv. $N \log N$

v. N^2

Question 3: Trace/Debug . . . . *11 marks*

(a) [2 marks] Point out any errors in the following statements.

Statement	**Error (if any)**
`float = 3.14*2;`	
`float stu marks = 40.5;`	
`FirstName = 'Donald';`	
`char n = '7';`	

(b) [2 marks] Consider the following method:

```
void mystery(int n) {
   if (n > 100) {
      std::cout<<n;
   } else {
      mystery(2 * n);
      std::cout<<", " + n;
   }
}
```

For each of the following calls, indicate the output that is produced by the method:

i. `mystery(60);`

ii. `mystery(41);`

. . . .

. . . .

. . . .

(c) [1$^{1}/_{2}$ marks] What is the output of the following :

```
void main() {
   int result = mystery2(348);
   std::cout<<"Result is : "<<result;
}

int mystery2(int num) {
   if (num < 10)
      return num;
  else
      return num%10 + mystery2(num/10);
}
```

. . . .

. . . .

. . . .

(d) [2 marks] Consider the following method, `mystery3`:

```
void mystery3(vector<int> a, vector<int> b) {
   for (int i = 0; i < a.length; i++) {
      a[i] += b[b.length - 1 - i];
   }
}
```

What are the values of the elements in array `x` after the following code executes?

```
Vector<int> x = {1, 3, 5, 7, 9};
Vector<int> y = {1, 4, 9, 16, 25};
mystery3(x, y);
```

....

....

(e) [1 mark] What does the following code fragment print?

```
String string1 = "hello";
String string2 = string1;
string1 = "world";
std::cout<<string1;
std::cout<<string2;
```

....

....

(f) [1 mark] What does the following code fragment print?

```
std::string s = "Hello World";
for (size_t i = 0; i < s.length(); ++i) {
s[i] = std::toupper(s[i]);
}
std::string sub = s.substr(6, 5);
std::cout << s << std::endl;
std::cout << sub << std::endl;
```

....

....

(g) [1½ marks] What does the following recursive function return?

```
string mystery4(String s) {
   int N = s.length();
   if (N <= 1) return s;
   String a = s.substr(0, N/2); String
   b = s.substr(N/2, N);
   return mystery4(b) + mystery4(a);
}
```

....

....

Question 4: Methods . . . . *5 marks*

(a) [$2\frac{1}{2}$ marks] Write a method called `isSorted` that given an array of integers returns **`true`** if array is sorted in ascending order and **`false`** otherwise.

(b) [$2\frac{1}{2}$ marks] Write a static method called `sumTo` that accepts an integer parameter *n* and returns a real number representing the sum of the first *n* reciprocals. In other words, `sumTo(n)` returns $(1 + 1/2 + 1/3 + 1/4 + \cdots + 1/n)$. For example, `sumTo(2)` shouldreturn `1.5`. The method should return `0.0` if it is passed the value ≤ 0.

Question 5: Arrays .... *5 marks*

Write a method called `contains` that accepts two arrays of integers `A` and `B` as parameters and that returns a boolean value indicating whether or not the sequence of elements in `B` appears in `A` (**`true`** for yes, **`false`** for no). The sequence must appear consecutively and in the same order. For example, consider the following arrays:

```
int list1[9] = {1, 6, 2, 1, 4, 1, 2, 1, 8};
int list2[3] = {1, 2, 1};
```

The call of `contains(list1, list2)` should return **`true`** because the sequence of values in `list2{1, 2, 1}` is contained in `list1` starting at index 5. If `list2` had stored the values `{2, 1, 2}`, the call of `contains(list1, list2)` would return **`false`**. Any two lists with identical elements are considered to contain each other. Every array contains the empty array, and the empty array does not contain any arrays other than the empty arrayitself.

Question 6: Recursion .... *5 marks*

Write a ***recursive*** method called `doubleDigits` that accepts an integer *n* as a parameter and returns the integer obtained by replacing every digit of *n* with two of that digit. For example, `doubleDigits(348)` should return `334488`. The call `doubleDigits(x)` for $x \leq 0$ should return `0`.