

NAME Solution

SID (last 4) \_\_\_\_\_

LOGIN \_\_\_\_\_

**CS 010 – Intro to Computer Science**

Winter - 2004

Mid-term exam – Friday 1/30 – total 100 points

Section 001

Time: 45 minutes

You may have on your desks ONLY this exam, a writing implement and an eraser.

You may use a separate sheet(s) of scratch paper. Make good use of this – don't write your answers in the assigned space until you are sure of them!

WRITE CLEARLY IN THE SPACE PROVIDED: illegible responses will not be graded

You will be required to show your student ID when you hand in your exam.

Be sure to read each problem carefully and follow the directions.

Problem 1	16	
Problem 2	12	
Problem 3	12	
Problem 4	18	
Problem 5	12	
Problem 6	12	
Problem 7	18	
<b>TOTAL</b>	<b>100</b>	

NAME \_\_\_\_\_

SID (last 4) \_\_\_\_\_

LOGIN \_\_\_\_\_

1. (16 pts - 2 each) True/False. Please write out "True" and "False". Do not just write T or F.

- a. There is a syntax error in the following output statement. `cout << "Hello\n" << "World\n" << endl;` *False*
- b. The following statement is a legal variable declaration. `char my_char = 'me';` *False*
- c. The following statement is **NOT** a legal variable declaration. `int my_int == 10;` *True*
- d. After the following two statements, x will hold the value 30.  
`int x;`  
`x = x + 30;` *False (not initialized)*
- e. After the following two statements, x will hold the value 1.0.  
`double x;`  
`x = 12 / 8;` *True (integer division)*
- f. The word `3rd_num` is a valid identifier (variable or function name). *False*
- g. The Boolean expression  $(x < y)$  is equivalent to the Boolean expression  $!(x > y)$ . *False  $(!(x >= y))$*
- h. Only one of the Boolean expressions on either side of an OR (`||`) operator must be true for the entire expression to be true. *True*

2. (12 pts - 3 each) Fill in the blank.

- a. What is a correct conditional statement (Boolean expression) to determine if x is **NOT** between 2 and 5?  
 *$(x < 2 \ || \ x > 5)$*
- b. An important step in designing a program is to first write an algorithm in pseudocode or English.
- c. The main difference between a do-while loop and a while loop is that the body of the while loop may never execute but the body of the do-while loop will always execute at least once.
- d. What is the logic error in the following statement?  
`fahrenheit = (9 / 5) * celsius + 32.0;`  
 *$9/5$  is integer division  
 $(9/5.0$  would work)*

LOGIN \_\_\_\_\_

3. (12 pts) Write an **if-else** statement that outputs "No jacket required!" if the variable *temperature* is greater than or equal to 75 and the variable *rainfall* is less than or equal to 0.2. Otherwise the **if-else** should output "Don't forget your jacket!"

```

if (temperature >= 75 && rainfall <= 0.2)
{
    cout << "No jacket required!";
}
else
{
    cout << "Don't forget your jacket!";
}

```

4. (18 pts - 6 each) To the right of each of the following code fragments, show its output.

a. 

```
int i = 0;
while (i <= 10)
{
    cout << i << " ";
    i += 2;
}
cout << i << endl;
```

 0 2 4 6 8 10 12

b. 

```
int i = 0;
do
{
    i += 2;
    cout << i << " ";
} while (i <= 10);
cout << i << endl;
```

 2 4 6 8 10 12 12

c. 

```
int x = 4, y = 7, z = 1;
if (x < y)
{
    z = z + x;
}
else
{
    z = z + y;
}
cout << z << endl;
```

 5

LOGIN \_\_\_\_\_

5. (12 pts) Write a **function declaration** and **function definition** that takes two arguments of type *char* and returns a value of type *int*. It should return a 1 if the two characters are the same character and a 0 if the characters are not the same.

```
int same-char(char var1, char var2);
```

```
int same-char(char var1, char var2)
{
    if (var1 == var2)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
```

6. (12 pts) Write a **function declaration** and **function definition** for a function that converts seconds to minutes. It should take one argument of type *int* that represents seconds and return a value of type *double* that is the equivalent value of minutes. So, if 30 is the argument passed in, 0.5 should be the return value.

```
double minutes(int seconds);
```

```
double minutes(int seconds)
{
    return seconds / 60.0;
}
```

NAME \_\_\_\_\_

SID (last 4) \_\_\_\_\_

LOGIN \_\_\_\_\_

7. (18 pts) Write a **program** that takes one positive integer as input from the user and outputs the sum of all the numbers between 0 and the user's input. So, if the user types in 5 the output should be 15 ( $1 + 2 + 3 + 4 + 5 = 15$ ).

Sample run: (user input in **bold**)

Enter a positive integer: **5**

The sum of all the numbers between 0 and 5 is 15

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

```
int main()
{
```

```
    int user_num, sum = 0, cnt = 0;
```

```
    cout << "Enter a positive integer: \n";
```

```
    cin >> user_num;
```

```
    while (cnt <= user_num)
```

```
    {
```

```
        sum += cnt;
```

```
        cnt++;
```

```
    }
```

```
    cout << "The sum of all the numbers between 0 and "
```

```
        << user_num
```

```
        << " is "
```

```
        << sum;
```

```
    return 0;
```

```
}
```

