

The American University in Cairo
Computer Science & Engineering Department
CSCE 106 "Fundamentals of Computer Science"

Instructor: Dr. Howaida Ismail	Final Exam	Spring 2013
Last Name :		ID:
First Name:		Form I Section No.: ()

EXAMINATION INSTRUCTIONS

- * Do not turn this page until asked to do so.
- * Exam time is **120** minutes.
- * Put the answers on the same question sheet, do not use any additional papers, even for scratch.
- * Write your name, ID, section no. in the indicated places.
- * Read the exam instructions.
- * Read the honesty policy.
- * Sign the following statement.

Academic Integrity Policy

Cheating in Exams is a violation of the honesty policy of AUC. Whispering, talking, looking at someone else's paper, or copying from any source is considered cheating. Any one who does any of these actions or her/his answers indicates that she/he did any of them, will receive a punishment ranging from zero in this exam to failing the course. If repeated, it may lead to dismissal from AUC.

I have read the honesty policy and exam instructions and I am presenting this exam as entirely my effort.

Signature: _____

DO NOT USE THIS SECTION

Question	Points	Grade
1	10	
2	20	
3	10	
4	15	
5	15	
6	10	
7	20	
Total	100	

Question 1 (10 points)

Tick only one possible answer for each of the following:

- 1) The Boolean expression $!(x > y)$ can be simplified into:
- a. $(x < y)$ ☐
 - b. $(x \geq y)$ ☐
 - c. True ☐
 - d. none of the above ☐
- 2) **Intellectual Property:**
- a. A form of protection that can be granted by the government for an invention; gives exclusive rights of an invention to its inventor for 20 years. ☐
 - b. Standards of moral conduct as they relate to computer use. ☐
 - c. The rights to which creators of original creative works (such as software, artistic, literary works, inventions, and more) are entitled. ☐
 - d. The legal right to sell, publish, or distribute an original software, artistic or literary work; is held by the creator of a work as soon as it exists in physical form. ☐
- 3) The value of $((x - 4) != 4) \&\& ((x - 4) == 4)$ is:
- a. $(x - 4)$ ☐
 - b. False ☐
 - c. $(x + 4)$ ☐
 - d. none of the above ☐
- 4) The **Compiler** is a software that:
- a. translates a high-level language program into machine language ☐
 - b. translates an assembly language program into machine language ☐
 - c. lets users display and view a web document ☐
 - d. None of the above ☐
- 5) Suppose that x is an int variable that contains the value 2 and y is an int variable that contains the value 5, the new value of x after the execution of $x += 2 * y$; will be
- a. 1 ☐
 - b. 12 ☐
 - c. 10 ☐
 - d. None of the above ☐
- 6) The **ALU** is the unit of computer responsible of:
- a. Storing programs and data permanently ☐
 - b. Performing arithmetic and logical operations ☐
 - c. Performing Input/Output operations ☐
 - d. None of the above ☐
- 7) The value of $((x \leq y) \parallel \text{True})$ is:
- b. $(x > y)$ ☐
 - c. $(x \leq y)$ ☐
 - d. False ☐
 - d. none of the above ☐
- 8) In C++, the statement: `if (1 == 1) cout << 1; else cout << 0;` will produce:
- c. 0 ☐
 - d. `( 1 != 1)` ☐
 - e. 1 ☐
 - d. none of the above ☐
- 9) $(110)_{10} + (110)_2$ is equivalent to:
- a. $(220)_{10}$ ☐
 - b. $(1110100)_2$ ☐
 - c. $(1100)_2$ ☐
 - d. None of the above ☐
- 10) A computer crime in which a person's social security number is taken is called:
- d. Hacking ☐
 - e. Identity theft ☐
 - f. E-harassment ☐
 - d. Social re-engineering ☐

Question 1 (20 points)

Show the output of each of the following program segments:

<pre>void drawLine(char); int func(int); void main () { drawLine('#'); cout << setw(3) << "k" << setw(8) << "func(k)" << endl; drawLine('-'); for (int k=0; k <= 5; k += 2) cout << setw(3) << k << setw(8) << func(k) << endl; drawLine('#'); } int func(int n) { int f = 1; for (int m=2; m <= n; m++) f = m * f; return f; } void drawLine(char ss) { for (int c = 1; c <= 10; c++) cout << ss; cout << endl; }</pre>	
<pre>float a = 3, b = 2, c = 1.5, x; int d = 4, e = 6.5, y; x = a * b - c + ++d / e++; y = a * (b - c) + e / --d; cout << " x = " << setw(4) << x << endl; cout << " y = " << setw(4) << y << endl; a = 3.3456; b = 5.9876; cout << setprecision(2); cout << setw(4) << a << setprecision(1) << setw(4) << b << endl; cout << setprecision(0) << setw(4) << a << setw(4) << b << endl;</pre>	
<pre>void swap (int&, int&); void main() { int a = 6.5, b = 5.3, c = 3.2; do { if (b < a) swap (a, b); if (c < b) swap (b, c); } while ((b < a) (c < b)); cout << "The Final value is:" << endl; cout << setw(4) << (b + c) / 2 << endl; } void swap (int& X, int& Y) { int T = X; X = Y; Y = T; }</pre>	

```

int n = 11001, w = 1, s = 0;
const int ten = 10;
cout << "The equivalent value of " << n << " = ";
do
{
    s = s + w * (n % ten);
    w = 2 * w;
    n = n / ten;
} while ( n != 0 );
cout << s << endl;

```

```

bool check ( int );
void main ()
{
    int AB [5] = { 12, 10, 101, 33, 330};
    for ( int index = 0; index < 5; index++)
        if ( check ( AB[index] ) )
            AB[index] = 1;
        else
            AB[index] = 0;
    for ( int r = 0; r < 5; r++)
        cout << setw(3) << AB[r];
    cout << endl;
    system ( "pause" );
}
bool check ( int n)
{
    const int ten = 10;
    bool flag = true;
    do
    {
        if ( n % ten == 0 )
            flag = false;
        else
            n = n / ten;
    } while ( ( n != 0 ) && ( flag ) );
    return flag;
}

```

Question 3 (10 points)

Draw the Flow Chart and the final output of the following program segment:

```
int a = 3.5, b = 5.6, c = 7.7;  
if ((a > b) && (a > c))  
    cout << "the result = " << a << endl;  
else  
    if (b > c)  
        cout << "The result = " << b << endl;  
    else  
        cout << "The result = " << c << endl;
```

Output

Flow Chart

Question 4 (15 points)

1. The following C++ function **rotate** takes three integer numbers over its parameters **a**, **b**, and **c** and then rotates them, such that **a** takes the value of **c**, **b** takes the value of **a**, and **c** takes the value of **b**. After being rotated, the three values should be returned back to the **main** function. Complete the following code of the function **rotate**:

```
..... rotate ( ..... a, ..... b, ..... c)
```

{

...../

...../

...../

...../

...../

}

2. Write a C++ function **exists** that takes a big positive integer number (**big**) and a single digit (**d**) from **0** to **9**. The function returns the number of occurrences of the digit (**d**) in the big positive number (**big**). (For example: if the function takes **27817** as a big integer number, and **7** as a single digit, then, it returns **2**, and if the function takes **345086** as a big integer number, and **9** as a single digit, then, it returns **0**.)

[illegible]

Question 5 (15 points)

Write only the C++ nested-loop to produce the following output:

```
@ @ @ @ @ @ @ @ @
1 2 3 4 5 6 7 8
& & & & & &
@ @ @ @ @
1 2 3 4 5
& & & &
@ @ @
1 2
&
```

Question 6 (10 points)

A Prime number is a positive integer number greater than 1 and divisible only by 1 and itself. The first prime numbers in the range of positive integers are 2, 3, 5, 7, etc.

The following modular C++ program **computes the sum** of all the prime numbers in the range of 1 to 1000. There are some missings (represented by dots) in the given program. Complete these missings such that the program could be compiled and run correctly.

The Program

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

.....

void main ()
{   int sum = .....

    for ( int c = ..... )

        if ( ..... )

            .....

    cout << "The Sum of Prime Numbers in the range 1 to 1000 = " << setw(4) << sum << endl;
    system ( "pause" );
}

bool prime ( ..... m )
{   bool flag = .....
    int s = 2;
    while ( ..... )

        If ( ..... )

            .....

        else

            .....

    return ( ..... );
}
```

Write a C++ modular program to generate a table showing the equivalent decimal value of a binary value. The table contains 20 rows at most. The program goes through the following steps:

1. The **main** function takes a positive integer number **n** representing the number of rows to be printed in the table. The program validates the entered value for **n** to accept only a value that is greater than **0** but not more than **20**.
2. The **main** function prints out the header of the table (see the given example below). Reads the binary numbers one by one. Each entered binary number is only combination of 0's and 1's. For each entered binary number, a function **convertBtoD** is called to take the binary number, compute its equivalent decimal value and return it back to the **main** function. The program prints a row with the two values; the binary number and its equivalent decimal value.
3. After printing **n** rows, the program prints a footer (see the example below).
4. You are advised to define and use appropriate functions to print the header and footer of the table.

Example: for **n = 5**, and binary numbers = **101, 1101, 111, 1111, and 10001**, the generated conversion table should look like the following one:

Binary Number	Decimal Value
101	5
1101	13
111	7
1111	15
10001	9

[illegible]