

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

Instructor:	Final Exam	Fall 2010
Last Name :		ID:
First Name:		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	25	
3	15	
4	15	
5	15	
6	20	
Total	100	

Question 1 (10 points)

Tick only one possible answer for each of the following:

- 1) 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 ☐
- 2) Summing up all even integers is a nonalgorithmic problem, because:
 - a. It's too difficult to compute ☐
 - b. It involves infinite number of operations without termination ☐
 - c. No machine can do it ☐
 - d. The algorithm to do it will very long ☐
- 3) The value of **(x - 4 != 4) && (x - 4 == 4)** is:
 - a. **x - 4** ☐
 - b. True ☐
 - c. **x + 4** ☐
 - d. None of the above ☐
- 4) $(11111111)_2$ is equivalent to:
 - a. $(128)_{10}$ ☐
 - b. $(127)_{10}$ ☐
 - c. $(255)_{10}$ ☐
 - d. None of the above ☐
- 5) Von Neumann architecture is:
 - a. A stored-program computer ☐
 - b. A web browser ☐
 - c. An architecture for RAM ☐
 - d. All the above ☐
- 6) **!((x == x) && (y > 10))** is equivalent to:
 - a. **True** ☐
 - b. **(x != x) && (y < 10)** ☐
 - c. **(x != x) || (y <= 10)** ☐
 - d. None of the above ☐
- 7) The RAM is used to:
 - a. To store fixed code to start up the computer ☐
 - b. To store programs and data that are to be executed ☐
 - c. To perform the arithmetic and logical operations ☐
 - d. none of the above ☐
- 8) In binary: $1011 + 11011 = ?$
 - a. 101000 ☐
 - b. 100110 ☐
 - c. 110110 ☐
 - d. 100101 ☐
- 9) **((x == x) || false)** is equivalent to
 - a. **x == x** ☐
 - b. **false** ☐
 - c. **x != x** ☐
 - d. None of the above ☐
- 10) **((x == x) && true)** is equivalent to
 - a. **x == x** ☐
 - b. **false** ☐
 - c. **true** ☐
 - d. None of the above ☐

Question 2 (25 points)

Show the output of each of the following program segments:

<pre>bool myfunc(int n); int main () { for (int k=2; k <= 30; k += 3) If (myfunc(k)) cout << setw(4) << k; cout << endl; } bool myfunc(int n) { bool flag = true; int c = 2; while ((c <= n/2) && (flag)) if (n % c == 0) flag = false; else c++; return flag; }</pre>	
<pre>int x = 1, y = 1, z; y = y * x++; cout << "x = " << setw(2) << x << "y = " << setw(2) << y << endl; x = 6; y = 2; z = --x / y++; y = z-- + x++; cout << "x = " << x << " y = " << y << " z = " << z << endl;</pre>	
<pre>float a = 2, b = 2, c = 1.5, x; int d = 5, e = 4.5, y, z; x = a * b - c + d / e; y = a * (b - c) + d / e; z = a * (b - (c + d - 0.5) / e); cout << "x = " << setw(4) << x << " y = " << setw(4) << y << " z = " << setw(4) << z << endl;</pre>	

Question 3 (15 points)

Write a C++ program that uses **only two nested loops** to produce the following output:

```
@ @ @ @ @ @ @ @ @  
 # # # # # # # # #  
   & & & & & & &  
     @ @ @ @ @ @ @ @  
       # # # # # # # #  
         & & & & & &  
           @ @ @  
             # #  
               &
```

[illegible]

Question 4 (15 points)

Write a C++ program that computes and outputs the volume of a cone and area of its base, given the diameter of its base and its height. The program should validate the user input to accept only values that are greater than 0 for both diameter and height. The formulas for computing the cone's base area and volume are:

Area of base = $\pi \times \text{Radius}^2$

Volume of cone = $\frac{1}{3} \times \pi \times \text{Radius}^2 \times \text{Height}$

Print both computed values to the nearest tenth. Be sure to use proper formatting and appropriate comments in your code. The output should be clearly labeled.

Show the three phases of software development: the analysis, design (draw a Flow Chart), and implementation.

The Analysis

[illegible]

The Flow Chart

This image shows a full page of a document template designed for handwritten notes or essays. It features approximately 28 evenly spaced, horizontal grey lines across the entire width of the page. The margins are consistent on all sides, providing ample space for writing. There are no other markings, text, or graphics present on the page.

[illegible]

Question 5 (15 points)

The ancient Greek Euclid developed a method for finding the Greatest Common Divisor (GCD) of two integers **A** and **B** (**A** greater than **B**). His method is:

1. If the remainder (**R**) of **A / B** is 0, then **B** is the **GCD**.
2. If it is not 0, then assign **B** to **A** and the remainder (**R**) to **B**.
3. Return to step 1 and repeat the process.

The following C++ program takes two integer numbers **N** and **M** and prints out their **GCD**. The program is structured as three functions: the **main**, **swap** and **GCD**. The **main** function takes two integer numbers **N** and **M** and if **N < M** it calls the function **swap** that will interchange the values of **N** and **M** (**N** is assigned to **M** and **M** is assigned to **N**). Then, the function **main** calls the function **GCD** that receives the values of **N** and **M** as **A** and **B**, performs Euclid's process (defined above) and returns the corresponding **GCD** value to the **main** function that prints it out.

The C++ program is given below but there are some missings (represented by dots). Complete these missings such that the program could be compiled and run correctly.

```
#include <iostream>
#include <iomanip>
using namespace std;
void swap(int&, int&);
.....
void main ()
{   int N, M;
    cin >> N >> M;
    if (N < M)
        .....

    cout << "The Greatest Common Divisor of " << setw(4) << N << " and " << setw(4) << M << " is " <<
    ..... << endl;
}

void swap (.....)
{
    int T = X;
    .....
    .....
}

..... GCD (int A, int B)
{ int R;
  R = A % B;
  while ( ..... )
  { A = B;
    .....
    R = .....
  }
  return .....;
}
```

Question 6 (20 points)

Write a complete modular C++ program that produces a table showing the temperature equivalents in degrees Fahrenheit and degrees in Celsius. The table consists of a header, set of rows and a footer (see the given example below). Let the user enter the starting, ending, and step values. Use the formula:

celsTemp = 5/9 (farenTemp – 32.00)

You are advised to define four functions: **main**, **header**, **footer**, and **convertToCelsius**. The **main** functions calls the **header** function to print the **header** of the table (as shown in the given example), takes from the user the starting, ending and step values (with enforcing the necessary validation that the ending value should be greater than the starting value), go over each Fahrenheit value in the range to call the function **convertToCelsius** that takes the Fahrenheit value and returns its equivalent Celsius value according to the formula given above. The **main** function prints the Fahrenheit value and its equivalent Celsius value in one row of the table. After printing the last row in the table, the **main** function calls the **footer** function that prints the footer (18 dashes).

- - - - -

Fahrenheit | Celsius

- - - - -

. . .

. . .

. . .

. . .

. . .

- - - - -

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Good Luck