

Name: _____

Section: _____

Print legibly!

COMPUTER SCIENCE 261

PROGRAMMING CONCEPTS

EXAM 1 – VERSION 1

Fall 2014

150 Points

Absolutely ***no electronic devices*** may be used during this exam.

1. No cell phones, computers, calculators, music players, headphones, etc.
2. Set your cell phone to be ***completely silent***, not vibrate. If it vibrates during the exam, it will be confiscated.
3. Place these and any other electronic devices in your backpack, ***out of immediate reach***.

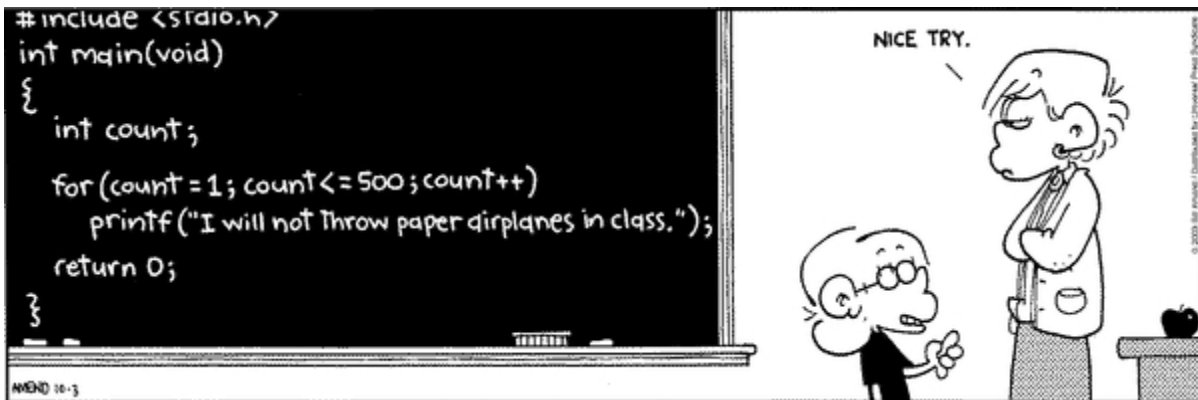
Failure to follow these instructions will result in a ***zero for the exam***.

You are expected to complete this exam ***in its entirety*** before leaving the exam room.

CLOSED BOOK – CLOSED NOTES

YOUR SCORE: _____ POINTS

A little humor to relax you ...



All of your answers must be written in clear, readable text. *If we cannot read it, it is not correct.*

1. 10 points. **True** or **False**. Write T or F to the left of each statement.

 T Pseudocode helps programmers develop their algorithm before implementing it with code.

 F Consider the statement `(expr1) || (expr2)`. Once `expr1` evaluates to 1, the compiler will continue to evaluate `expr2`.

 T You cannot write an *else* statement without a corresponding *if*.

 F You can not use a conditional statement (such as *if-else* or *switch*) inside the code block of another conditional statement.

 F If I know how many times I want a loop to run, the best choice is a *while* loop.

 T The concept of abstraction allows users of functions to view the functions as a ‘black box’.

 F Several math functions, e.g., `pow()` and `abs()`, are standard functions and do not require the user to include a header file.

 T A digital photo (or music file) is a stream of bits, nothing more.

 F `fin` and `fout` are the standard file input and output streams.

 F The *continue* statement will quit a loop, i.e., stop the loop from repeating further.

2. 9 points. Given the following variable declarations, write the output of each **cout** statement on the line to the right. There are ***no errors*** in this code, but Visual Studio may show warnings for some. Show the output that would be produced if warnings are ignored and the program is executed.

```
int a = 17, b = 3;
double x(5.0);
int y(0);
```

```
y = a / b;
cout << y;
```

_____ **5** _____

```
x = a / b;
cout << x;
```

_____ **5** _____

```
y = a % b;
cout << y;
```

_____ **2** _____

```
y = b % a;
cout << y;
```

_____ **3** _____

```
y = (double) a / b;
cout << y;
```

_____ **5** _____

```
x = (double) a / b;
cout << x;
```

_____ **5.6667** _____

3. 3 points. Given the following variable declarations, write **true** or **false** on the line to the right to indicate the result of each logical statement.

```
int a = -23, b = 7, c = 15, x = 0, y = 32;
```

```
( a + 1 - b ) != ( y / 3 )
```

_____ **true** _____

```
( a < b || b < c ) && ( x < a )
```

_____ **false** _____

```
( a < b && c < b ) || ( x < y && x < b )
```

_____ **true** _____

4. 15 points. Fill in each of the 13 blanks.

In order to create a different pseudorandom sequence we give a different integer called a **seed** as an argument to the `srand` function.

What is the output of the following statements: **a is 5 and x is 5**

```
int a = 4;
int x = ++a;
cout << "a is " << a << " and x is " << x;
```

What is the output of the following statements: **a is 5 and x is 4**

```
int a = 4;
int x = a++;
cout << "a is " << a << " and x is " << x;
```

The most appropriate type of variable for y below is **bool**.

```
y = (!(a > 5.0))
```

The **break** statement is used to end the evaluation of further choices in a switch statement.

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

An else statement **does NOT** [does | does not] have a boolean expression associated with.

Valid whitespace characters are **spaces**, **tabs**, and **newlines**.

We create a **class** to define a new data type.

The term **scope** describes which areas of a program can access specific variables.

Including a function **prototype** above `main()` allows a programmer to place the function definition below `main()`.

5. 7 points. Multiple choice.

A **const** keyword when used in a variable declaration:

- a) means that the variable can only be assigned data with digits.
- b) means the variable name should be written in camel case.
- c) means the data assigned to the variable cannot be modified.**
- d) all of the above.
- e) none of the above.

What is the result of this code snippet?

```
for (int i=0; i <= 9; ++i)
{
    cout << i << ", ";
}
cout << i;
```

- a) 0,1,2,3,4,5,6,7,8,9
- b) 0,1,2,3,4,5,6,7,8,9,10
- c) Error**
- d) Nothing (no output)

Circle the functions that are available to an input file stream:

- a) eof()**
- b) put()
- c) open()**
- d) unsuccessful()
- e) clear()**

6. 6 points. What is written on standard output after the following C++ snippet of code is executed.

```
int option = 3;
switch (option)
{
    case 1:
        cout << "Red\n";
        break;
    case 2:
        cout << "Green\n";
    case 3:
        cout << "Blue\n";
    case 4:
    case 5:
        cout << "Cyan\n";
    default:
        cout << "Magenta\n";
}
```

PUT OUTPUT IN BOX:

Blue
Cyan
Magenta

7. 10 points. Write a snippet of code that will read in user integer input until the inputted value is between 10 and 20 (inclusively), and will then output that number divided by 4 (with a decimal value, if not evenly divisible). If the user inputs a value outside the range, the user should be notified of the input error, and prompted to ask for input again. Notes: Don't worry about invalid inputs (i.e. putting a letter in instead of a number). Don't worry about #includes or defining a main function, but any variables you use that are not in a library (like cin/cout) must be declared.

```
int input;
do
{
    cout << "Enter an integer between 10 and 20" << endl;
    cin >> input;
} while ((input < 10) || (input > 20));

cout << "The value is " << input / 4.0 << endl;
```

8. 15 points. Write a snippet of code to check if a given integer is prime. First check that the user entered a number greater than one. If not, display an error message and exit. If yes, output whether the number provided is prime. Hint: To check if a number n is prime, it should not be divisible by any number between 2 and $n/2$. Don't worry about #includes or defining a main function, but any variables you use that are not in a library (like cin/cout) must be declared.

```
int input;
cout << "Enter a number to check whether it is prime" << endl;
cin >> input;
if (input <= 1)
{
    cerr << "error on input; exiting" << endl;
    return 1;
}

bool prime = 1;
for (int i = 2; i <= (input / 2); i++)
    if ((input % i) == 0)
        prime = 0;

if (prime == 0)
    cout << "input is NOT prime" << endl;
else
    cout << "input is prime" << endl;
```

9. 15 points. Ima Poorstudent has written the following program, which you can see has several mistakes. Fix these mistakes by updating the code so that it (1) will compile and (2) makes sense. All input/output should be from standard I/O. Ignore compiler warnings.

```

include <fstream>      → #include<iostream>
using namespace std;

int myFunc(int x);

int main()
{
    char charA;
    cout << "Enter a character, then press enter \n"; → \n";
    cin << char;      → cin >> charA;

    for (j = 4; j < 8; j++); → for (int j=4; j<8; j++)
        cout << "j is /t" << j << endl; → \t

    cout << "Function is " << myFunc(7.4) << endl;

    return 0;
}

void myFunc(int x); → int myFunc(int x)
{
    fout << "I'm in the function with x = " << x << endl;
    → cout
    if ( x = 0 ) → if (x == 0)
        cout << "x is equal to zero " << endl;

    return x;
}

```

10. 10 points. What is the output of the following code segment?

```
int countIt = 2;

int total = 0;

cout << "count    total" << endl;
while( countIt <= 7 )
{
    cout << countIt << "    " << total << endl;

    total += countIt;

    ++countIt;
}
cout << "Final total is " << total << endl;
```

PUT OUTPUT IN BOX:

count	total
2	0
3	2
4	5
5	9
6	14
7	20
Final total is 27	

11. 10 points. What is the output of the following code segment?

```
void theFunc(int x, int y, int z);

int main()
{
    double a(9.2), b(8.5), c(7.8);
    cout << a << " " << b << " " << c << endl;
    theFunc(a, b, c);
    cout << a << " " << b << " " << c << endl;
    return 0;
}

void theFunc(int x, int y, int z)
{
    cout << x << " " << y << " " << z << endl;
    x = 3;
    y = 4;
    z = 5;
    cout << x << " " << y << " " << z << endl;
}
```

PUT OUTPUT IN BOX:

9.2	8.5	7.8
9	8	7
3	4	5
9.2	8.5	7.8

12. 20 points. Complete the program below such that:

- a) a function prototype and definition is added for MPG, which has no return and no parameters
- b) a function prototype and definition is added for doMPG, which returns a double and has two double parameters (miles and gas)
- c) function main calls function MPG
- d) function MPG asks the user to enter miles traveled and gas (in gallons) used
- e) function MPG calls doMPG
- f) function doMPG calculates fuel efficiency (miles per gallon)
- g) function doMPG returns the value calculated
- h) function MPG displays the result returned

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

```
void MPG();
```

```
double doMPG(double miles, double gas);
```

```
int main()
```

```
{
    MPG();
    return 0;
}
```

```
void MPG()
```

```
{
    double miles, gas;
    cout << "Miles traveled? ";
    cin >> miles;
    cout << "Gas used (in gallons)? ";
    cin >> gas;
    cout << doMPG(miles, gas) << " miles per gallon" << endl;
}
```

```
double doMPG(double miles, double gas)
```

```
{
    return (miles / gas);
}
```

13. 20 points. In this problem, you are to write short C++ snippets of code (no main function is required). Assume the proper libraries have been included.
- Write a **single line** C++ statement that declares and opens the file “myDataIn.txt” for an input stream named “myInput”.
 - Write a snippet of code that tests whether the input stream from (a) was successfully opened. If the file did not open properly, quit the program.
 - Write a **single line** C++ statement that reads one line of data from the file “myDataIn.txt” using “myInput”. The line of data consists of one integer accountNum and a floating point number currentBalance. (You may assume that the variables are declared properly.)
 - Write a **single line** C++ statement that declares and opens the file “myDataOut.txt” for an output stream named “myOutput”.
 - Write a **single line** C++ statement that writes one line of data to the file “myDataOut.txt” using “myOutput”. The line of data consists of the integer and floating point number from part (c). The only difference is you should subtract 50 from currentBalance.

a) **ifstream myInput(“myDataIn.txt”);**

b) **if myInput.fail()**
{
cerr << “File error; exiting” << endl;
return 1;
}

c) **myInput >> accountNum >> currentBalance;**

d) **ofstream myOutput(“myDataOut.txt”);**

e) **myOutput << accountNum << “\t” << (currentBalance – 50) << endl;**