**Homework: Primitive Data Types and Variables**

**Problem 1. Declare Variables**

* Declare five variables choosing for each of them the most appropriate of the types byte, sbyte, short, ushort, int, uint, long, ulong to represent the following values: 52130, -115, 4825932, 97, -10000.
* Choose a large enough type for each number to ensure it will fit in it. Try to compile the code.

Submit the source code of your Visual Studio project as part of your homework submission.

**Problem 2. Float or Double?**

* Which of the following values can be assigned to a variable of type float and which to a variable of type double: 34.567839023, 12.345, 8923.1234857, 3456.091?
* Write a program to assign the numbers in variables and print them to ensure no precision is lost.

**Problem 3. Variable in Hexadecimal Format**

* Declare an integer variable and assign it with the value 254 in hexadecimal format (0x##).
* Use Windows Calculator to find its hexadecimal representation.
* Print the variable and ensure that the result is 254.

**Problem 4. Unicode Character**

* Declare a character variable and assign it with the symbol that has Unicode code 42 (decimal) using the \u00XX syntax, and then print it.

Hint: first, use the Windows Calculator to find the hexadecimal representation of 42. The output should be \*.

**Problem 5. Boolean Variable**

* Declare a Boolean variable called isFemale and assign an appropriate value corresponding to your gender.
* Print it on the console.

**Problem 6. Strings and Objects**

* Declare two string variables and assign them with Hello and World.
* Declare an object variable and assign it with the concatenation of the first two variables (mind adding an interval between).
* Declare a third string variable and initialize it with the value of the object variable (you should perform type casting).

**Problem 7. Quotes in Strings**

* Declare two string variables and assign them with following value: The "use" of quotations causes difficulties.
* Do the above in two different ways - with and without using quoted strings.
* Print the variables to ensure that their value was correctly defined.

**Problem 8. Isosceles Triangle**

* Write a program that prints an isosceles triangle of 9 copyright symbols ©, something like this:

©

© ©

© ©

© © © ©

*Note: The © symbol may be displayed incorrectly at the console so you may need to change the console character encoding to UTF-8 and assign a Unicode-friendly font in the console.*

*Note: Under old versions of Windows the © symbol may still be displayed incorrectly, regardless of how much effort you put to fix it.*

**Problem 9. Exchange Variable Values**

* Declare two integer variables a and b and assign them with 5 and 10 and after that exchange their values by using some programming logic.
* Print the variable values before and after the exchange.

**Problem 10. Employee Data**

A marketing company wants to keep record of its employees. Each record would have the following characteristics:

* First name
* Last name
* Age (0...100)
* Gender (m or f)
* Personal ID number (e.g. 8306112507)
* Unique employee number (27560000…27569999)

Declare the variables needed to keep the information for a single employee using appropriate primitive data types. Use descriptive names. Print the data at the console.

**Problem 11. Bank Account Data**

* A bank account has a holder name (first name, middle name and last name), available amount of money (balance), bank name, IBAN, 3 credit card numbers associated with the account.
* Declare the variables needed to keep the information for a single bank account using the appropriate data types and descriptive names.

**Problem 12. Null Values Arithmetic**

* Create a program that assigns null values to an integer and to a double variable.
* Try to print these variables at the console.
* Try to add some number or the null literal to these variables and print the result.

**Problem 13.\* Comparing Floats**

* Write a program that safely compares floating-point numbers (double) with precision eps = 0.000001.

*Note: Two floating-point numbers a and b cannot be compared directly by a == b because of the nature of the floating-point arithmetic. Therefore, we assume two numbers are equal if they are more closely to each other than a fixed constant eps.*

Examples:

| **Number a** | **Number b** | **Equal (with precision eps=0.000001)** | **Explanation** |
| --- | --- | --- | --- |
| 5.3 | 6.01 | false | The difference of 0.71 is too big (> eps) |
| 5.00000001 | 5.00000003 | true | The difference 0.00000002 < eps |
| 5.00000005 | 5.00000001 | true | The difference 0.00000004 < eps |
| -0.0000007 | 0.00000007 | true | The difference 0.00000077 < eps |
| -4.999999 | -4.999998 | false | Border case. The difference 0.000001 == eps. We consider the numbers are different. |
| 4.999999 | 4.999998 | false | Border case. The difference 0.000001 == eps. We consider the numbers are different. |

**Problem 14.\* Print the ASCII Table**

* Find online more information about ASCII (American Standard Code for Information Interchange) and write a program that prints the entire ASCII table of characters on the console (characters from 0 to 255).

*Note: Some characters have a special purpose and will not be displayed as expected. You may skip them or display them differently.*