



## Problem Set 5: Names and Data Types

Due: 11:59 PM, Monday, October 28, 2019

This problem set consists of several relatively short programming tasks and other questions. You may work alone or with a partner on this assignment.

---

### Getting Set Up

You will receive an email with the link to follow to set up your GitHub repository for this problem set (`ps5-yourgitname`). If you are working on this problem set with a partner, **one** member of the group should follow the link to set up the repository on GitHub. Your problem set partner will receive a subsequent email with a link to follow that will grant them the rights to clone the repository and commit and push changes to the origin on GitHub.

By 4:00 PM, Tuesday, October 22, 2019, all repositories must be created and if working in a group, the partner who did not create the repository must have write access to the repository. Your name(s) must be in the repository by this deadline as well.

---

### Languages on noreaster

This assignment will ask you to write programs in four languages: C, C++, C#, and Java. Please make sure all programs run on noreaster. In case you never knew or have forgotten, here is how to compile and run a program in each of these languages (using examples from the late penalty calculators in the shared area).

- C using `gcc`

To compile:

```
gcc -o late late.c -lm
```

Note that `-lm` is needed only if linking with math library functions.

To run:

```
./late
```

- C++ using `g++`

To compile:

```
g++ -o late late.cc
```

To run:

```
./late
```

- C# using Mono

To compile:

```
mcs Late.cs
```

To run:

```
mono Late.exe
```

- Java using JDK command line

To compile:

```
javac Late.java
```

To run:

```
java Late
```

---

## Questions and Programs

All programs required are to be treated as “Practice Programs” in terms of collaboration and grading. You are welcome to discuss them with your classmates, and I will grade only on correctness, not documentation, style, etc. Of course, you should still make sure your name is in every file you submit! Unless otherwise specified, all programs should be included in your submission and you should make sure they run on noreaster. Written responses may be in a PDF file included in your repository or in the repository’s README.md file.

### ? Question 1:

Compute the address of the `[5][2]` element in a  $10 \times 10$  2-dimensional array of integers (4 bytes each) called `IntArray` where the initial address for the array starts at 00004400 for C++ and Fortran. Show your work. (4 points, 2 each)

### ? Question 2:

Do Programming Exercise 5 at the end of Sebesta Chapter 5, p. 233. Note that in order to make some of your programs compile, you might need an instance or global variable also named `x` outside of your function or method’s scope, and you’ll want to sprinkle in some appropriate printouts so you can better see what’s happening. And don’t forget to explain what you found in addition to submitting your three programs. (12 points)

**? Question 3:**

Do Programming Exercise 6 at the end of Sebesta Chapter 5, p. 233. To clarify, the programs should demonstrate whether a variable declared in the initialization part of the for loop declaration (not the body of the loop) is visible outside the loop. Be sure to state the scoping rules you discovered for each of these languages. (5 points)

**? Question 4:**

Do Programming Exercise 7 at the end of Sebesta Chapter 5, p. 233. You could time your code by instrumenting it with timers (see, for example, C's `gettimeofday` function), or by timing the execution of your entire program with the Unix `time` command. The `time` command can be added before any Unix command line and an extra line of output will appear after your program's output that includes useful information including the wall-clock and CPU time taken to run your program. And don't forget the last sentence of the problem: "Explain the results." (8 points)

**? Question 5:**

Given a two's complement, Little Endian architecture, show the contents of memory needed to store the 32 bit integer -17, located starting at address 00CCCC00. (2 points)

Address	Contents
00CCCC00	
00CCCC01	
00CCCC02	
00CCCC03	

**? Question 6:**

Programming Exercise 7 at the end of Sebesta Chapter 6, p. 298. Include your C program in your submission. Include your actual timing results as well as a brief analysis in your response. (8 points)

**? Question 7:**

Give the IEEE 754 32-bit representation of the number 16.25. Do not worry about endianness. (2 points)

Sign Bit	Exponent	Mantissa

**? Question 8:**

Give the BCD representation for the number 2615 in binary and in hexadecimal. (2 points)

**? Question 9:**

Summarize the essential differences between the mechanisms to handle character strings in C and Java. (3 points)

We saw in the example of Pascal subranges that with the `fpc` compiler on noreaster, an assignment of a variable to a constant outside of the subrange generates a warning at compile time, but that values outside the range are still allowed.

**? Question 10:**

Describe what the language's run-time system would need to do to enforce ranges of computed values at run time. For example, to generate an error from the statement below in the Pascal subranges program. (4 points)

```
Minute := Minute * 3;
```

**? Question 11:**

Draw a picture of the memory allocated for a  $4 \times 4$  two-dimensional array for a rectangular allocation (true two-dimensional array) and a jagged allocation (array of arrays). (3 points)

**? Question 12:**

Write a short Python program that creates and populates an array, then takes a slice through the array that takes every third element, starting with the second element (*i.e.*, the element at subscript 1). (3 points)

---

**Submission**

Commit and push!

---

**Grading**

This assignment will be graded out of 55 points.

Feature	Value	Score
Question 1: computing addresses	4	
Question 2: Programming Exercise 5 programs	9	
Question 2: Programming Exercise 5 explanation	3	
Question 3: Programming Exercise 6 programs	3	
Question 3: Programming Exercise 6 results	2	
Question 4: Ch. 5 Programming Exercise 7 program	6	
Question 4: Ch. 5 Programming Exercise 7 explanation	2	
Question 5: 2's complement	2	
Question 6: Ch. 6 Programming Exercise 7 program	5	
Question 6: Ch. 6 Programming Exercise 7 results and explanation	3	
Question 7: IEEE 754	2	
Question 8: BCD	2	
Question 9: Strings	3	
Question 10: Pascal ranges	4	
Question 11: 2D Array	2	
Question 12: Python Lists	3	
Total	55	