SIENA*college*
Computer Science

Computer Science 340
Programming Languages
Siena College
Fall 2023

Problem Set 5: Names and Binding

Due: 11:59 PM, Monday, October 30, 2023

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

In Canvas, you will find a link to follow to set up your GitHub repository, which will be named `names-probset-yourgitname`, for this problem set. Only one member of the group should follow the link to set up the repository on GitHub, then others should request a link to be granted write access.

All GitHub repositories must be created with all group members having write access and all group member names specified in the `README.md` file by 11:59 PM, Wednesday, October 25, 2023. This applies to those who choose to work alone 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 repository).

- 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: Problem Set Question 6 at the end of Sebesta Chapter 5, p. 228. (6 points)

Question 2: Problem Set Question 7 at the end of Sebesta Chapter 5, p. 229. (6 points)

Question 3: Problem Set Question 8 at the end of Sebesta Chapter 5, p. 229-30. (6 points)

Question 4: Problem Set Question 9 at the end of Sebesta Chapter 5, p. 230. (6 points)

Question 5: Problem Set Question 10 at the end of Sebesta Chapter 5, p. 231. (6 points)

Question 6: 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 7: 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. (8 points)

Question 8: 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." (10 points)

Submission

Commit and push!

Grading

This assignment will be graded out of 60 points.

Feature	Value	Score
Q1: PS 6	6	
Q2: PS 7	6	
Q3: PS 8	6	
Q4: PS 9	6	
Q5: PS 10	6	
Q6: PE 5	12	
Q7: PE 6	8	
Q8: PE 7	10	
Total	60	