

Example Solutions for Programming Languages G22.2110 Summer 2007 hw06

Assigned Th 6/29/2007, due Th 7/5/2007 at 1pm.

These are example solutions. Please keep in mind that often, there is not just one correct solution to a question. If you come up with different answers to the homework, then it may be that both your answers and these answers here are correct. Of course, these answers here may also contain mistakes; if you spot some, please let us know so we can correct them.

Reading Assignments

- For lecture on 6/28/2007: Scott 7.1.0-7.1.3, 7.2.0-7.2.2;
Wheeler 1-6, 8-9 (<http://www.adahome.com/Tutorials/Lovelace/master.htm>)
 - For lecture on 7/5/2007: Scott 9.1, 9.4.3;
Sections from Java tutorial (<http://java.sun.com/docs/books/tutorial/java/TOC.html>)
about Classes, Objects, More on Classes, Interfaces, Inheritance, Strings, Packages
-

Homework Assignments

1. Structural equivalence (16 = 4 + 4 + 4 + 4 points)
Consider the following definition of structural equivalence:

Two types T_1 and T_2 are equivalent if they allow the same sequence of field selections, array subscripts, and pointer dereferences to be applied, and if a sequence of such operations applied on type T_1 and ending in a primitive type always ends in the same primitive type when applied to type T_2 .

- 1a. (4 points) Are `struct A` and `struct B` structurally equivalent? Explain briefly.

```
struct A { double d; struct A* pa; };  
struct B { double d; struct B* pb; };
```

Example solutions

No, `struct A` and `struct B` are not structurally equivalent, because the field names `pa` and `pb` differ. They violate the path-based definition of structural equivalence because for example path `x->pa->d` is possible in `struct A`, but not in `struct B`, since `struct B` has no field named `pa`.

- 1b. (4 points) Are struct C and struct D structurally equivalent? Explain briefly.

```
struct C { struct C* p; int i[5]; };
struct D { struct D* p; int i[5]; };
```

Example solutions

Yes, struct C and struct D are structurally equivalent, because they both allow paths of the form $x \rightarrow p \rightarrow \dots \rightarrow p \rightarrow i[0..4]$ ending in a primitive int.

- 1c. (4 points) Are struct E and struct F structurally equivalent? Explain briefly.

```
struct E { struct C* q; struct F* r; };
struct F { struct D* q; struct E* r; };
```

Example solutions

Yes, struct E and struct F are structurally equivalent, because both allow paths to int of the form $x(->r)^* \rightarrow q(->p)^* \rightarrow i[0|1|2|3|4]$, where r and p occur zero or more times, and q and i occur exactly once.

- 1d. (4 points) Are struct G and struct H structurally equivalent? Explain briefly.

```
struct G { struct A a; };
struct H { struct A a; };
```

Example solutions

Yes, because both allow paths of the form $x.a \rightarrow pa \rightarrow \dots \rightarrow pa \rightarrow d$.

2. Ada (8 = 0 + 2 + 3 + 3 points)

The code at the following URLs defines an Ada package `Int_Stacks` and a driver for it:

<http://www.cs.nyu.edu/courses/summer07/G22.2110-001/hw06-ada-driv.txt>

<http://www.cs.nyu.edu/courses/summer07/G22.2110-001/hw06-ada-spec.txt>

<http://www.cs.nyu.edu/courses/summer07/G22.2110-001/hw06-ada-body.txt>

- 2a. (0 points) Download the files, and rename them to `driver.adb`, `int_stacks.ads`, and `int_stacks.adb`, respectively. Compile and run the program, and test it with some inputs.
- 2b. (2 points) The “with” clause of the `Int_Stacks` module body imports a system library. What is the URL for this library in the Ada RM?

Example solutions

<http://www.adahome.com/rm95/rm9x-13-11-02.html>

- 2c. (3 points) Remove the “use `Ada.Text_IO`” clause from the driver, then change the code until it compiles and runs again without that clause. What did you need to change?

Example solutions

Access module members with dot notation instead of simple identifiers.

--before--	--after--
Get_Line(Line, Characters)	Ada.Text_IO.Get_Line(Line, Characters)
Put_Line("--> void")	Ada.Text_IO.Put_Line("--> void")

and similarly for two more `Put_Line` calls.

- 2d. (3 points) What would happen if the programmer forgot the line `Free_Int_Array(S.Data);` in the implementation of `Push`?

Example solutions

There would be a memory leak, since the old data array would not be reclaimed from the heap, even though the pointer to it gets overwritten with a pointer to the new data array. In this example, that would not cause a serious problem. But in general, avoiding memory leaks can prevent performance degradation and crashes.

3. Ada (6 = 3 + 3 points)

While you write the Ada code for answering Question 4 below, you will probably get some error messages. Describe two error messages using the following format:

- Code: *a very short piece of code that triggers the error*
- Symptom: *the error message itself*
- Cause: *an explanation for what triggered the error message*
- Solution: *how to fix the code to prevent the error*

Example solutions

```
- first error
- Code: if S(I) == C then
- Symptom: == should be =
- Cause: Ada uses = for equality comparison, unlike C-like languages,
  which use ==.
- Solution: if S(I) = C then
- second error
- Code: Put_Line("CountOccurrences(\"hello\", 'l')");
- Symptom: binary operator expected
- Cause: Ada uses double-doublequote (") instead of
  backslash-doublequote (\") for literal doublequotes in strings.
- Solution: Put_Line("CountOccurrences(""hello"", 'l')");
```

4. Ada (20 = 5 + 5 + 5 + 5 points)

Write Ada-95 programs exercising the fundamental features.

- 4a. I/O (5 points)

Write a program that prompts the user for his or her name, reads the name from input, then politely greets the user by name. You can assume that the user response does not exceed 100 characters. Here is an example interactive session:

```
What is your name?
Bob
Hello, Bob, nice to meet you!
```

Example solutions

```
with Ada.Text_IO; use Ada.Text_IO;
procedure Main is
  N : constant := 100;
  Line : String(1 .. N);
  Characters : Natural;
begin
  Put_Line("What is your name?");
  Get_Line(Line, Characters);
  Put_Line("Hello, " & Line(1 .. Characters) & ", nice to meet you!");
end Main;
```

4b. Libraries (5 points)

Write a program that uses Ada-95 library functions to compute $\sqrt{2}$, $\sin(3.5)$, and $e^{2.5}$, and then prints the results like this (don't worry if the numbers are displayed in a slightly different format):

```
square root of 2.0:      1.41421E+00
sine of 3.5:            -3.50783E-01
e to the power of 2.5:  1.21825E+01
```

Example solutions

```
with Ada.Text_IO; use Ada.Text_IO;
with Ada.Numerics.Elementary_Functions; use Ada.Numerics.Elementary_Functions;
procedure Main is
  Sqrt2 : Float := Sqrt(2.0);
  Sin35 : Float := Sin(3.5);
  Exp25 : Float := Exp(2.5);
begin
  Put_Line("square root of 2.0: " & Float'Image(Sqrt2));
  Put_Line("sine of 3.5: " & Float'Image(Sin35));
  Put_Line("e to the power of 2.5: " & Float'Image(Exp25));
end Main;
```

4c. Types (5 points)

The following code creates a variable C with the character value 'Z', and then prints a description and the value of the variable:

```
C : Character := 'Z';
Put_Line("name C, type Character, value " & Character'Image(C));
```

Extend this program by creating and printing more variables of different types. Your program should produce the following output:

```
name C, type Character, value 'Z'
name F, type Float, value 3.14100E+00
name I, type Integer, value 42
name IA, type array of Integer, value ( 1, 4, 9, 16)
```

name S, type String, value hello

Example solutions

```
with Ada.Text_IO; use Ada.Text_IO;
procedure Main is
  C : Character := 'Z';
  F : Float := 3.141;
  I : Integer := 42;
  IA : array(1 .. 4) of Integer := (1, 4, 9, 16);
  S : String := "hello";
begin
  Put_Line("name C, type Character, value " & Character'Image(C));
  Put_Line("name F, type Float, value " & Float'Image(F));
  Put_Line("name I, type Integer, value " & Integer'Image(I));
  Put("name IA, type array of Integer, value (");
  for J in IA'Range loop
    if IA'First /= J then
      Put(", ");
    end if;
    Put(Integer'Image(IA(J)));
  end loop;
  Put_Line(")");
  Put_Line("name S, type String, value " & S);
end Main;
```

4d. Control flow (5 points)

Write an Ada function `CountOccurrences` that takes two parameters, a string and a character, and returns the number of occurrences of the character in the string. For example, `CountOccurrences("hello", 'l')` should return 2.

Example solutions

```
with Ada.Text_IO; use Ada.Text_IO;
procedure Main is
  function CountOccurrences(S : String; C : Character) return Integer is
    Result : Integer := 0;
  begin
    for I in S'Range loop
      if S(I) = C then
        Result := Result + 1;
      end if;
    end loop;
    return Result;
  end CountOccurrences;
begin
  Put_Line("CountOccurrences(" & "hello" & ", 'l') -> "
    & Integer'Image(CountOccurrences("hello", 'l')));
```

```
end Main;
```

5. Java (0 points)

Start teaching yourself Java by doing the following:

- 5a. If possible, find peers (other students who want to learn Java together with you) and gurus (people who already know Java, whom you can ask questions when you get stuck).
- 5b. Make sure you have access to a Java 1.5 compiler (javac) and virtual machine (java). These and other tools come bundled in a JDK (Java developer kit) for J2SE (Java 2 standard edition) from vendors like Sun, IBM, or BEA. Ask your guru if you are having problems with this step.
- 5c. Go to <http://java.sun.com/docs/books/tutorial/java/TOC.html> and read at least the following sections: Classes, Objects, More on Classes, Interfaces, Inheritance, Strings, Packages. Along the way, try things out with the Java compiler and virtual machine from Step 5b. You might want to skim earlier sections as needed. You do not need to read the following sections: Nested Classes, Enum Types, Annotations, Numbers, Characters, Generics.
- 5d. Familiarize yourself with the structure of the online Java documentation, so you can find information quickly when you need it. In particular, find the URLs for the Java Language Specification and the J2SE API specification.

<http://www.cs.nyu.edu/courses/summer07/G22.2110-001/hw06-example-solutions.pdf>

Total points: 50.