



University of Arizona, Department of Computer Science
CSc 372 — Assignment 7 — Due noon, Mon Nov 28 — 10%

Christian Collberg
November 18, 2005

1 Introduction

The purpose of this assignment is to get started writing Icon programs.

We will be grading on correctness and style. Every procedure should be adequately formatted and documented.

Note that the code we're writing in this assignment will be used in the next!

2 Point ADT

We're going to write an abstract data type POINT that manipulates points in the plane. Here is the template for the module:

```
record POINT(x,y)

# Return a random point POINT(x,y) where x
# is in [1..W] and y in [1..L]
procedure point_random(W, L)
end

# Compute the normal of a point a=POINT(x,y),
# i.e. the square root of  $x^2 + y^2$ .
procedure point_normal(a)
end

# Return a list of n random points POINT(x,y) where x
# is in [1..W] and y in [1..L]
procedure point_randomList(n, W, L)
end

# p is either a POINT or a list of POINTs.
# Return the string "(x,y)" for the point POINT(x,y).
# If p is a list of POINTs, return [(x1,y1),(x2,y2),...].
procedure point_2string(p)
end

# a and b are POINTs.
```

```

# Return the smaller of two points a=POINT(ax,ay)
# and b=POINT(bx,by). a is smaller than b if
# ax<bx or ax=bx and ay<by
procedure point_MIN(a,b)
end

# This is a generic routine for adding two
# entities a and b where either can be a
# * POINT
# * numeric
# * list of POINTs
# Two POINTs add to a POINT, two lists of POINTs
# to a list of POINTs, two numerics to a numeric.
# To add a point a=POINT(x,y) and a number N,
# means returning a new point POINT(x+N,y+N).
# To add a list L=[POINT(x1,y1),POINT(x2,y2),...]
# and a point P=POINT(a,b) means returning
# a new list consisting of the elements of L
# each + P, i.e. [POINT(x1+a,y1+b),POINT(x2+a,y2+b),...].
procedure point_ADD(a,b)
end

# Same as point_ADD, but for subtraction.
procedure point_SUB(a,b)
end

# Same as point_ADD, but for multiplication
procedure point_MUL(a,b)
end

# p is a POINT or a list of POINTs.
# open up a window of width W and length L
# (centered around the point (0,0)) and
# draw a small circle for each point along
# the coordinates of the point.
procedure point_draw(p, W, L)
  WOpen("size=" || W || ", " || L, "dx=" || W/2, "dy=" || L/2) | stop("can't open window")
  Font("mono,bold,14")
  EraseArea(-W/2,-L/2,W,L)

  # more here!

  DrawCircle(x, y, 5)      # Draw the circle at position (x,y)
  DrawString(x-5, y-5, point_2string(v)) # Write "(x,y)" next to the circle

  WDone()
end

```

What's notable about this ADT is that many of the procedures (such as `point_ADD`) can operate on a variety of datatypes. `point_ADD`, for example, can operate on any combination of POINTs, numbers, and lists of POINTs. Icon's weak typing and built-in run-time typing functions (`type()`, `numeric`) makes this easy to

do.

Here's a simple test program:

```
link point
link graphics

procedure main()
  W := 500
  L := 500
  p1 := point_random(W,L)
  write("p1=", point_2string(p1))
  p2 := point_random(W,L)
  write("p2=", point_2string(p2))

  l1 := point_randomList(5,W,L)
  write("l1=", point_2string(l1))
  l2 := point_randomList(5,W,L)
  write("l2=", point_2string(l2))

  a1 := point_ADD(p1,p2)
  write("a1=", point_2string(a1))

  a2 := point_ADD(p1,l1)
  write("a2=", point_2string(a2))

  a3 := point_ADD(l1,p1)
  write("a3=", point_2string(a3))

  a4 := point_ADD(p1,5)
  write("a4=", point_2string(a4))

  a5 := point_ADD(l1,5)
  write("a5=", point_2string(a5))

  a6 := point_MIN(p1,p2)
  write("The MIN of ", point_2string(p1), " and ", point_2string(p2),
    " is ", point_2string(a6))

  a7 := point_normal(p1)
  write("The normal of ", point_2string(p1), " is ", a7)

  point_draw(l1,W,L)
end
```

It can be compiled and run with the commands

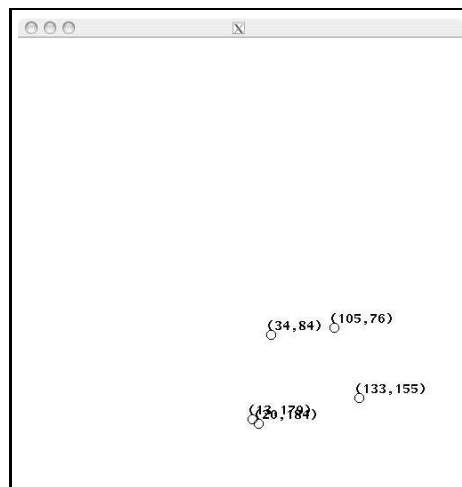
```
> icont -c point.icn
> icont test_point.icn
> test_point
```

```

p1=(53,103)
p2=(79,128)
l1=[(105,76),(20,184),(13,179),(34,84),(133,155)]
l2=[(144,177),(184,59),(212,79),(89,225),(211,3)]
a1=(132,231)
a2=[(158,179),(73,287),(66,282),(87,187),(186,258)]
a3=[(158,179),(73,287),(66,282),(87,187),(186,258)]
a4=(58,108)
a5=[(110,81),(25,189),(18,184),(39,89),(138,160)]
The MIN of (53,103) and (79,128) is (53,103)
The normal of (53,103) is 115.8360911

```

The call to `point_draw(l1,W,L)` will open up a window and draw each point in the plane. Most of this procedure has been given to you, all you have to do is to insert the code that draws a particular point, or goes through a list of points and prints all of them. Depending on your operating system, the result should look something like this:



Note: We will test `point_ADD` but not `point_SUB` and `point_MUL`. However, you should still make sure that they work!

Note: The `point.icn` template and `test_point.icn` are available on the course web site.

Note: You should do more extensive testing than is done in `test_point.icn`.

3 Icon functions

Here is a list of Icon operations that you may find useful for this assignment:

`list(n)`: Create a list consisting of length `n`. Each list element is `null`.

`list(n,e)`: Create a list consisting of `n` copies of `e`.

`type(n)`: Return the type of `n` as a string.

`numeric(n)`: Return a numeric representation of `n`, if possible, otherwise fail.

`?n`: Return a random number between 1 and `n`.

`*L`: Return the length of a list `L`.

`every i := 1 to n do {...}`: An Icon for-loop.

`a||b`: String concatenation.

`a|||b`: List concatenation.

`L[i]`: List indexing.

4 Submission and Assessment

The deadline for this assignment is noon, Mon Nov 28. It is worth 10% of your final grade.

You should submit the assignment electronically using the Unix command

```
turnin cs372.7 point.icn.
```

Don't show your code to anyone, don't read anyone else's code, don't discuss the details of your code with anyone. If you need help with the assignment see the instructor or the TA.