CMPSCI 119
Fall 2013
Monday, November 4, 2013
Midterm #2 Solution
Professor William T. Verts
20 Points – What is printed out by each of the following statements?

A. \[ \text{for } x \text{ in } \text{range}(3): \text{ printNow}(x) \]
   \[
   0 \ 1 \ 2
   \]

B. \[ \text{for } x \text{ in } \text{range}(3,12): \text{ printNow}(x) \]
   \[
   3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11
   \]

C. \[ \text{for } x \text{ in } \text{range}(3,12,2): \text{ printNow}(x) \]
   \[
   3 \ 5 \ 7 \ 9 \ 11
   \]

D. \[ \text{for } x \text{ in } \text{range}(0,5,1): \text{ printNow}(x) \]
   \[
   0 \ 1 \ 2 \ 3 \ 4
   \]

E. \[ \text{for } x \text{ in } \text{range}(\text{len("Frog")}): \text{ printNow}(x) \]
   \[
   0 \ 1 \ 2 \ 3
   \]

F. \[ \text{for } x \text{ in } "\text{Frog}": \text{ printNow}(x) \]
   \[
   "F" \ "r" \ "o" \ "g"
   \]

G. \[ \text{for } x \text{ in } [12,5,8]: \text{ printNow}(x) \]
   \[
   12 \ 5 \ 8
   \]

H. \[ \text{for } x \text{ in } [2]: \text{ printNow}(x) \]
   \[
   2
   \]

I. \[ \text{for } x \text{ in } [3.4, \ 6.9, \ 2.1]: \text{ printNow}(x) \]
   \[
   3.4 \ 6.9 \ 2.1
   \]

J. \[ \text{for } x \text{ in } [4, "\text{Q}", [4,5], 7.0]: \text{ printNow}(x) \]
   \[
   4 \ "\text{Q}" \ [4,5] \ 7.0
   \]

Assign 2 points per problem. Remove 1 point for getting some portion of the answer incorrect; remove 2 points if the entire answer is incorrect.
20 Points – What is printed out when `Main()` is called?

```python
def Whee(W):
    printNow(W)
    return

def Glop(X,Y):
    Whee(range(Y,X))
    return

def Junk(W,X):
    Glop(X,W)
    return

def Main():
    Glop(7,3)
    Junk(7,3)
    return
```

Answer:

```
[3,4,5,6]
[]
```

Here’s the calling sequence:

```
Main()
    Glop(7,3)
        Whee(range(3,7))
            printNow([3,4,5,6])  \[3,4,5,6] \text{printed}\n    Junk(7,3)
        Glop(3,7)
            Whee(range(7,3))
                printNow([])  \[\] \text{printed}\n```

Scoring: 10 points per answer. Give partial credit as appropriate.
20 Points – Using the 3D axes below and orthographic projection, draw a line from 3D location \(<5,1,3>\) to \(<-5,4,2>\) and also from \(<-3,-5,0>\) to \(<4,-3,3>\) (consider the tic marks to have a 1-unit separation):

Solution shown in red lines. Green and blue lines show projections from XZ plane. 10 points per red line – partial credit for “close” points.

20 Points – Point \(P\) is a list of two integers \([Xc,Yc]\) indicating the center coordinates of a diamond, as shown. Finish the \texttt{addCenteredDiamond} function below to draw the diamond (just the outline, not a filled diamond) centered at point \(P\) with the given radius and color. Your solution must use only JES functions.

```python
def addCenteredDiamond(Canvas, P, Radius, color):
    addLine(Canvas, P[0], P[1] - Radius, P[0] - Radius, P[1], color)
    addLine(Canvas, P[0] - Radius, P[1], P[0], P[1] + Radius, color)
    addLine(Canvas, P[0], P[1] + Radius, P[0] + Radius, P[1], color)
    addLine(Canvas, P[0] + Radius, P[1], P[0], P[1] - Radius, color)
    return
```

Assign 8 points for using four \texttt{addLine} commands, then 3 points per \texttt{addLine} command. Expected use is \(P[_]\pm\text{Radius}\) for all coordinates (no penalty for using \(R\) instead of \(\text{Radius}\), as long as a statement appears that assigns \(R = \text{Radius}\) in order to simplify the code). Remove points for syntax errors, up to 3 per statement. For other solutions use judgment as to quality of approach, syntax, etc.
20 Points – In the image at right of a train car, grid lines are 10 pixels apart and do not appear in the final drawing. You are given two functions for drawing circles and rectangles with one color for the interior (FillColor) and one for outlines (OutlineColor). A train car as shown is a blue car body plus two “trucks” that are cyan boxes each containing a pair of radius 20 white wheels, and four white “windows” each containing two 30×30 cyan window panes. The outline color in every case is black. The reference point for drawing each object (the train car, each window, and each truck) is in its upper left corner. Complete the functions below to draw a proper train car.

```python
def addCircle (Canvas,Xc,Yc,R,FillColor,OutlineColor):
    addOvalFilled(Canvas,Xc-R,Yc-R,2*R+1,2*R+1,FillColor)
    addOval(Canvas,Xc-R,Yc-R,2*R+1,2*R+1,OutlineColor)
    return

def addRectangle (Canvas,X,Y,W,H,FillColor,OutlineColor):
    addRectFilled(Canvas,X,Y,W,H,FillColor)
    addRect(Canvas,X,Y,W,H,OutlineColor)
    return

def addWindow (Canvas,X,Y):
    addRectangle(Canvas, X,Y,90,50,white,black)
    addRectangle(Canvas, X+10,Y+10,30,30,cyan,black)
    addRectangle(Canvas, X+50,Y+10,30,30,cyan,black)
    return

def addTruck (Canvas,X,Y):
    addCircle(Canvas, X+30,Y+40,20,white,black)
    addCircle(Canvas, X+80,Y+40,20,white,black)
    addRectangle(Canvas, X,Y,110,40,cyan,black)
    return

def addTrainCar (Canvas,X,Y):
    addRectangle(Canvas, X,Y,410,100,blue,black)
    addWindow(Canvas, X+10,Y+10)
    addWindow(Canvas, X+110,Y+10)
    addWindow(Canvas, X+210,Y+10)
    addWindow(Canvas, X+310,Y+10)
    addTruck(Canvas, X+20,Y+80)
    addTruck(Canvas, X+280,Y+80)
    return
```

There are 13 lines: 2 points each for the 7 addRectangle and addCircle commands, 1 point each for the 6 addWindow and addTruck commands. Remove ½ point per syntax/value error up to the limit allowed for each line.