December, 2024


**Figure 1, Smiling Face**

## The Problem:

Our traditional methods used in mathematics education often overlook opportunities to engage students in coding skills that will have a lasting impact in the future. We can improve our teaching methods by developing an understanding of math concepts for practical use for a society that is increasingly software driven.

December, 2024

Traditional grade school and high school math instructors can easily develop future coders by introducing coding as a computer lab exercise in the classroom.

The purpose for the math exercises is often not known by the students. Classes are mostly an exercise in memorizing formulas and procedures. The process is like training an artist to mix paints to attain various colors in the spectrum, without teaching them to paint on the canvas. CANVAS is an acronym for Computers Algebra Numbers Visual Aids *for* Science (CANVAS) . Coding CANVAS is the practice of using the computer screen as an artist uses a canvas. The goal is to utilize graphics (visual methods) in mathematics to assist in demonstrating applied math and computer science.

## The Solution:

The modern computer is a tool that is severely under-utilized in the classroom. The burden of doing mathematical computation and discovery of math concepts should be placed upon computers. Proper use of the computer will take students further down the educational road than current teaching methods afford. The computer's potential as a teaching tool is similar to paradigm shift caused by the advent of the automobile when horses and buggies, walking, streetcars, and bicycles were the dominant means for transportation. During this time period many questioned the benefits of a "horseless carriage" because it appeared to be an expensive toy. Similarly, the slower, more laborious hand teaching methods prevail, while computer aided instruction is considered a crutch that will hamper or hinder fundamental skills. Computer augmented training in math classes will fill a void that is now facing employers. Many technical industries are understaffed because insufficient numbers of students have the math and programming skills critical to this modern era. Computer augmented training will remove much of the burden placed on teachers to convey mathematical concepts to their students.

## The Approach:

Suppose that the subject of the day in a high school math class is to understand the mathematics of a circle with an offset. Most teachers using traditional methods will write the mathematical expression on the white board by hand then they will explain the variables for the points on the circle, the variable for the radius, and the variable for the offset. Next the instructor will attempt to explain how the variables change the circle's appearance. This is done by many iterations going from the changes in math expression, then to whiteboard to show the effect with multiple graphical plots of the expression. Many students will get lost in the iterations, variables and graphs. Furthermore, most students would not care about the math exercise because it seems pointless, except they need to get a good grade in the class.

As educators we are training the next generation of designers and developers. Necessity is the mother of invention. Development starts with need or necessity for the task at hand. Math should start with

establishing a need for the subject that is about to be taught. The purpose and the concept should be presented first. The development process has five steps:

1) establishing the need and the requirements,

2) the plan of action,

3) the preparation by the gathering of parts to implement the plans

4) the test if the requirements were satisfied (iterations will occur here).

5) Presentation of the solution and a summary of what was done.

How would this same exercise differ if a computer were used to in a classroom? Our instructor would define a task to use circles for, say, drawing a face. (See Figure 1, Smiling Face).

For grade schools, our instructor would introduce the number grid showing horizontal and vertical coordinates. Our instructor could place the facial features on the drawing, one at a time on the number grid, to show the importance of thoroughness and accuracy in math and computer science. We demonstrate this approach below using Python computer code.

The high school instructor would plan to use math expressions to draw the outline of the head with a circle. Likewise, the eyes and nose could be drawn using full circles with a smaller diameters and offsets. The mouth and ears can be done with partial circles, and differing diameters and offsets. Drawing the face on the board by hand could show this. The instructor will use the computer to draw the face using math expressions for a circle to render each of the facial features that were drawn by hand. Then the teacher reveals the common circle math expression to show what was changed to get the desired result. Variables can be changed to show the trends. The face can then be redrawn on the computer to help students see the cause and effect.

Without the computer, an instructor would have to work much harder to convey the concepts, and to keep the student's attention. Each student could then be required to manipulate the face using the computer to give another desired appearance. With success, the teacher then trains students in the use of a circle for more generalized cases. In summary, computers can be used in the classroom to illustrate mathematical principles more effectively by exposing students to pre-written source code and by allowing students to manipulate the code to see the cause and effect of changes in variables. At the same time, the science of learning to translate mathematical expressions to instructions understood by computers is a valuable skill set to develop for our students of tomorrow.

December, 2024

## The Test Cases:

How can one conduct the above exercise with students as a test case?

Instructors need only a modest amount of experience in math to participate. There are five test cases below: three Python examples, one Java example and one VBA example . Python language is available for free at https://python.org.

A valuable way to provide computer resources for students to use is to convert used computers to Ubuntu OS then to install Python language on these machines. Unbuntu is an open source OS that is free for download.

The first three approaches use Python coding language to place features of a face on a drawing using a number grid. We share three source code programs using tkinter and "graphics.py" libraries for Python.

The second coding example uses the tkinter graphics library to place the location of circles using a box that covers the outermost boundary of the circumference. This is the most difficult method of object placement.

The first coding example uses the graphics.py library tools from John Zelle to improve upon the tkinker graphics methods. The circles are located using the center and its radius. This is an improvement in object placement over the second coding method.

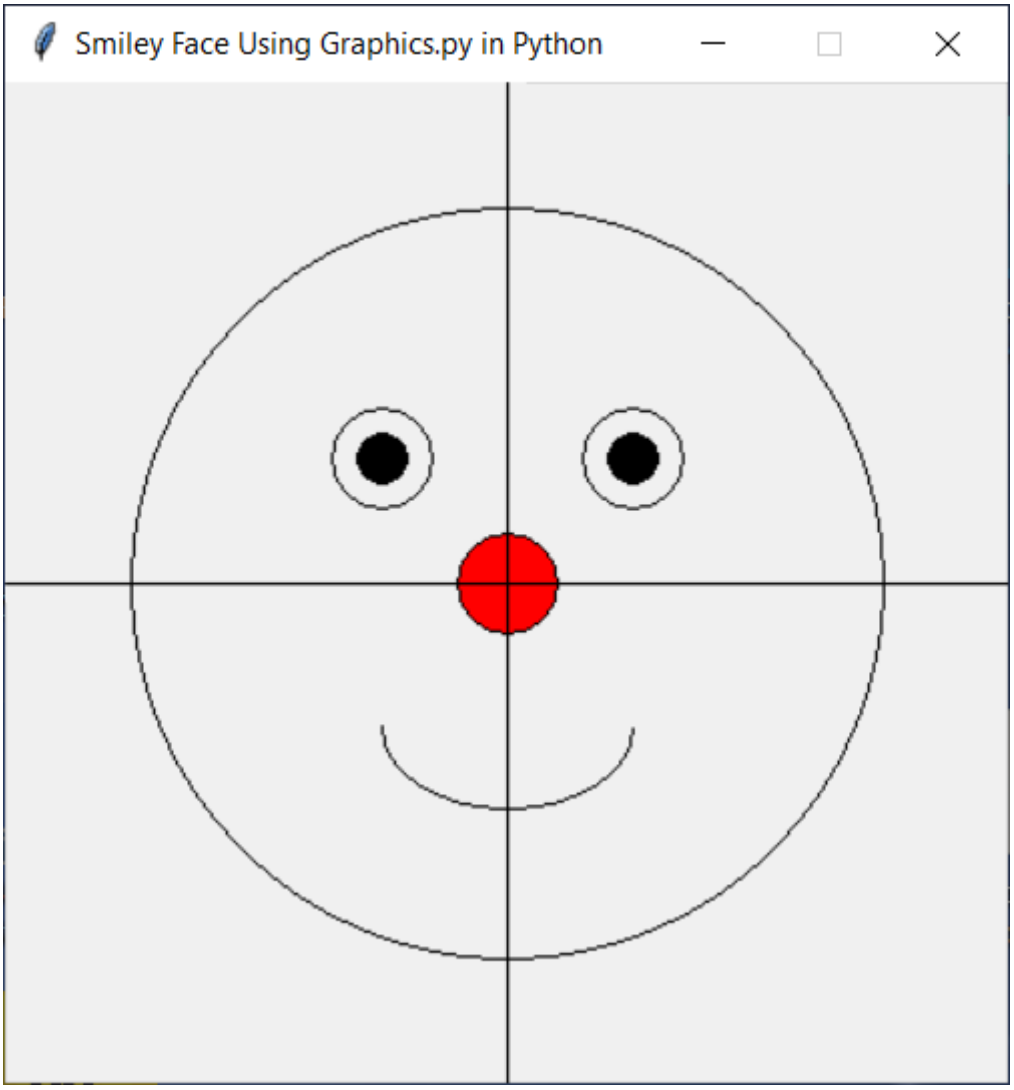
The third approach uses the graphics.py tools for the placement of the objects along with a re-mapping of the coordinates to make placement of the facial features easier. This is called a coordinate transformation. The window number grid ranges from -10,-10 in the lower left hand corner to 10,10 in the upper right hand corner. This is the easiest of the three Python test cases presented.

Please Google "graphics py" to obtain the Python graphics library of John Zelle.

Other test cases will be added in the future.

December, 2024

**Two Test cases for grade school**




**Python #1 Source Code Beginning:**

```
from graphics import *

import tkinter as tk

def main():

    win = GraphWin("Smiley Face Using Graphics.py in Python", 400, 400) #create a window for drawing

    c = Circle(Point(200,200), 20) #define nose circle
```

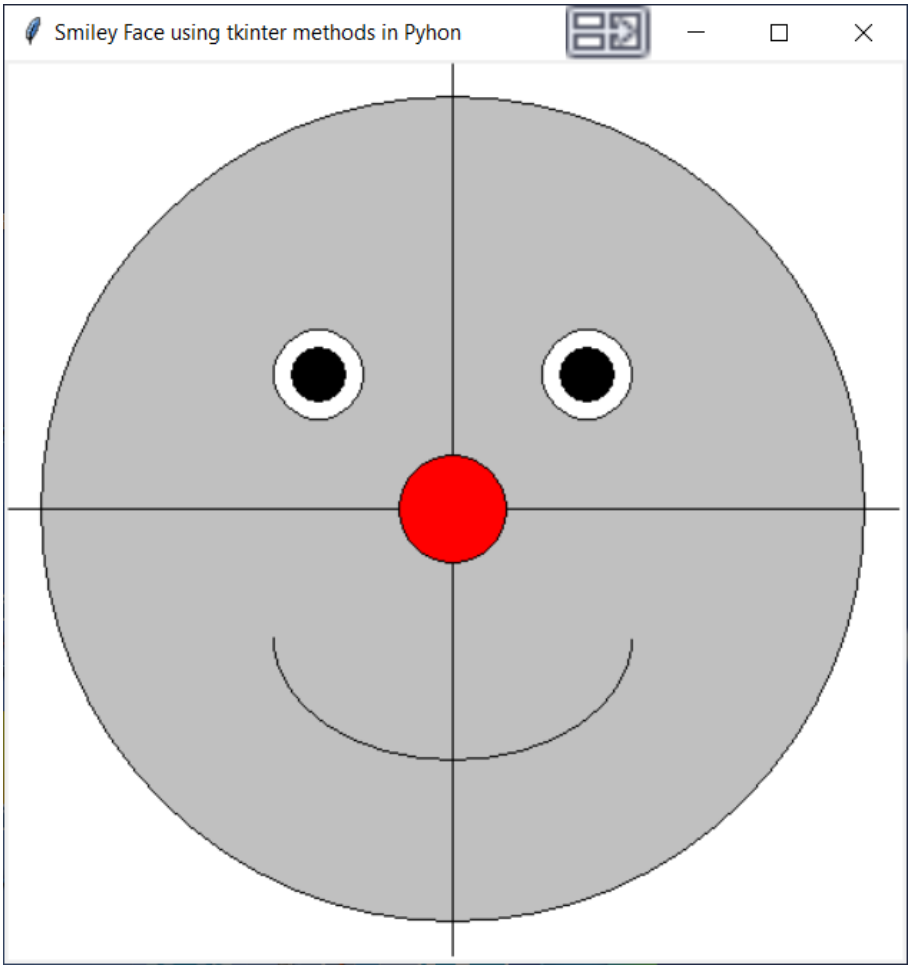
```
c1 = Circle( Point(150,150),20) #define left eye circle

c11= Circle( Point(150,150), 10) #define left eyeball

c2= Circle(Point(250,150),20) #define right eye circle

c21=Circle(Point(250,150), 10) #define right eye ball

c3= Circle(Point(200,200), 150) #define head circle

#root=tk.Tk()

#canvas = tk.Canvas(root, width=200, height=200)

#canvas.pack()

#Create an arc for the smile

win.create_arc(150,225,250,290, start=0, extent=-180, style=tk.ARC)

#root.mainloop

c.setFill("red") #fill nose with color

c11.setFill("black") #fill left eyball

c21.setFill("black") #fill right eyeball

pt1= Point(200,0) # define vertical line upper position

pt2 = Point(200,400) #define vertical line lower position

pt3= Point(0,200)#define horizontal line leftside

pt4= Point(400,200) #define horizontal line right side

d = Line(pt1,pt2) #define vertical line in window

e= Line(pt3,pt4) #define horizontal line in window

c.draw(win) #draw nose

c1.draw(win) #draw left eye

c11.draw(win) #draw left eyeball

c2.draw(win) #draw right eye
```

December, 2024

```
    c21.draw(win)#draw left eyeball
    c3.draw(win) #draw head
    d.draw(win) #draw vertical line
    e.draw(win) #draw horizontal line
    win.getMouse() # pause for click in window
    win.close() #close window after mouse click
main()
```

**Python #1 Source Code End:**

December, 2024




**Python #2 Source Code Beginning:**

```
from tkinter import *
from tkinter import messagebox
top = Tk()
top.title("Smiley Face using tkinter methods in Pyhon")
```

December, 2024

```
C = Canvas(top, bg="white", height=500, width=500)

head = C.create_oval(20,20,480,480,fill='silver')#draw head

vline = C.create_line(250,0,250,500,fill='black') # draw vertical line

hline = C.create_line(0,250,500,250,fill='black') # draw horizontal line

nose= C.create_oval(220,220,280,280, fill='red') #draw nose

l_eye= C.create_oval(150,150, 200,200, fill= 'white')# draw left eye

l_eyeball= C.create_oval(160,160,190,190, fill= 'black')# draw left eyeball

r_eye= C.create_oval(300,150, 350,200, fill= 'white')#draw right eye

r_eyeball= C.create_oval(310,160, 340,190, fill= 'black')#draw right eyeball

# draw smile

smile= C.create_arc(150,255,350,390, start=0, extent=-180, style=ARC)


C.pack()

top.mainloop()
```

**Python #2 Source Code End**

December, 2024


**Python #3 Source Code Beginning:**

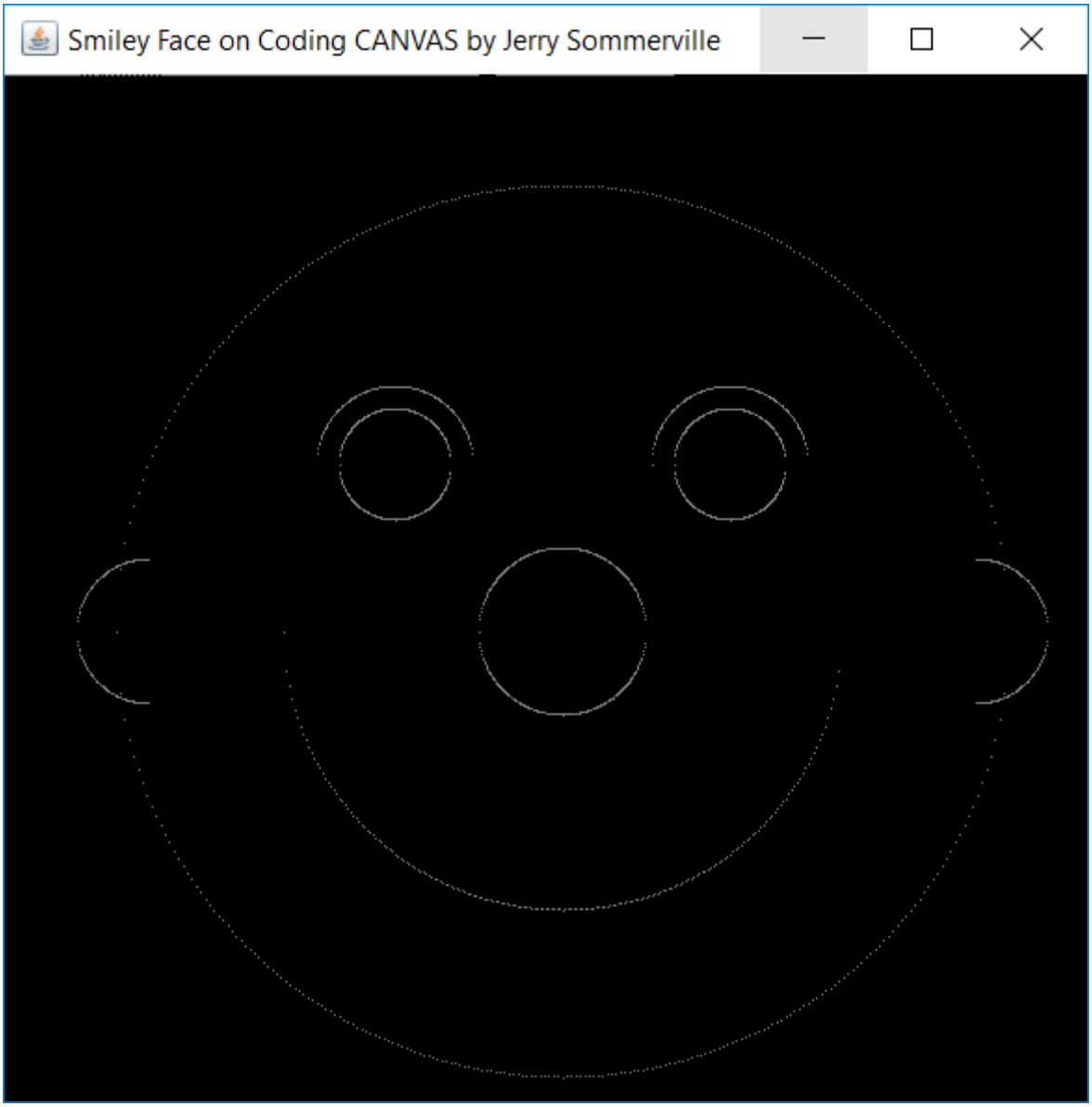
```
from graphics import *
import tkinter as tk
def main():
    win = GraphWin("Coord. Transform using graphics.py", 400, 400) #create a window for drawing
    win.setCoords(-10,-10,10,10) #re-map coordinates in the window
    head= Circle(Point(0,0),8)#define head
    head.setFill("tan")#fill face with color
```

```
    head.draw(win)#draw head
    nose = Circle(Point(0,0), 1.2)#define nose
    nose.setFill("red")#fill nose with color
    nose.draw(win)#draw nose in window
    reye= Circle(Point(3,3), 1)#define right eye
    reye.setFill("white")
    reye.draw(win)#draw right eye
    reyeball= Circle(Point(3,3), 0.5)# define right eyeball
    reyeball.setFill("black")#fill right eyeball with color
    reyeball.draw(win)#draw right eyeball
    leye= Circle(Point(-3,3), 1) #define left eye
    leye.setFill("white")
    leye.draw(win) #draw left eye
    leyeball= Circle(Point(-3,3), 0.5)#define left eyeball
    leyeball.setFill("black")#fill left eyeball color
    leyeball.draw(win)#draw left eyeball
    low_corner= Transform(400,400,-10,-10,10,10).screen(-4,-4)
    up_corner=Transform(400,400,-10,-10,10,10).screen(4,0)
    #draw smile
    win.create_arc(low_corner[0], low_corner[1],up_corner[0], up_corner[1], start=0, extent=-180,
style=tk.ARC)
    win.getMouse() # pause for click in window
    win.close() #close window after mouse click
main()
```

**Python #3 Source Code End**

**Java Example for High School**

December, 2024




The first exercise will use Java language. The figure below is the output for the Java Source Code that follows the image below. The source code develops places smiley face into window using JFrame. To display image below, use NotePad app in Windows to save file and run Java using command prompt. Java programs with "main" module will not display image using "NetBeans" IDE.

I summarize the approach and include excerpt of the actual source code below:

```
face_object(g, 8.0, 5, 0.0,0.0 ); // head
```

December, 2024

```
face_object( g, 1.5 , 5, 0.0, 0.0);// nose
face_object( g, 1.0, 5, 3,3);// right eye
face_object( g, 1.4, 1, 3,3);// right eye brow

face_object(g, 1.0, 5, -3,3); //left eye
face_object( g, 1.4, 1, -3,3);// left eye brow

face_object(g, 5.0,2,0,0); // Smile
face_object(g, 1.3,3,7.4,0); // right ear
face_object(g, 1.3,4,-7.4,0); //left ear
```

I made a "java method" called **"face_object"** that draws a circle or a part of a circle.

All that I do is change the circle's diameter and/ or its offset [x,y] (circle's distance away from the origin). The origin of the picture [0,0] is at the center of the nose. The left side is noted by a negative number for x. The right side is noted by a positive number for x. The same is true for y coordinate. The upper half is noted by a positive value for y. The bottom half is noted by a negative number for y. The full image scale is from -10 to +10 for x coordinate and the y coordinate.

Circle's radius is number next to "g" in parentheses.

Circle's offset [x,y] is the last two numbers in the parentheses

The type of circle such as the upper half (eye brow), bottom half(smile) , left side(ear), right side(ear), or full circle(head) is determined by the third item (that ranges from 0 to 5) in the parentheses.

**The numbers are clumsy, but the picture makes sense intuitively. This is why graphics and coding can help in math.**

The image and code shown is for "front-end" developers, so to speak. Front end developers use the code but they don't design it.

Modern day coding is done the same way. There is the front-end developer (WordPress), "window dressing" for the end users and there is the back-end developer for "the main code" (Database or application software) then there is the full stack developer who does both.

http://blog.udacity.com/2014/ 12/front-end-vs-back-end-vs- full-stack-web-developers.html

Eventually a front-end developer can become a back end developer with a little more knowledge.

December, 2024

Since Java is in such high demand, creating an opportunity to learn to Java and structured programming is the primary goal of the proposal by using a spoonful of sugar to make "the medicine" go down:

http://pypl.github.io/PYPL.html

http://www.codingdojo.com/blog/9-most-in-demand-programming-languages-of-2017/

http://www.business2community.com/tech-gadgets/top-20-popular-programming-languages-2017-01791470#g5rA3bhw4UV08oAE.97

**Java Source Code Beginning:**

```
// DrawFace.java by Jerry Sommerville

import java.awt.*;

import java.awt.event.*;

import java.awt.geom.Line2D;

import javax.swing.JApplet;

import javax.swing.JFrame;

import java.lang.*;


   public class DrawFace extends JApplet {

            public void init() {

             setBackground(Color.white);

            setForeground(Color.white);

            }

            public void paint(Graphics g) {

            Graphics2D g2 = (Graphics2D) g;
```

```
        //g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

        g2.setPaint(Color.red);


    face_object(g, 8.0, 5, 0.0,0.0 ); // head

    face_object( g, 1.5 , 5, 0.0, 0.0);// nose

    face_object( g, 1.0, 5, 3,3);// right eye

    face_object( g, 1.4, 1, 3,3);// right eye brow


    face_object(g, 1.0, 5, -3,3); //left eye

    face_object( g, 1.4, 1, -3,3);// left eye brow


    face_object(g, 5.0,2,0,0); // Smile

    face_object(g, 1.3,3,7.4,0); // right ear

    face_object(g, 1.3,4,-7.4,0); //left ear


        //g2.draw(new Line2D.Double(x, y, 400, 400));

        // g2.drawString("Line", x, 250);

        }

         public static void main(String s[]) {

        JFrame f = new JFrame("Smiley Face on Coding CANVAS by Jerry Sommerville");

        f.addWindowListener(new WindowAdapter() {

         public void windowClosing(WindowEvent e) {

         System.exit(0);

     }

        });

        JApplet applet = new DrawFace();
```

```
        f.getContentPane().add("Center", applet);
        applet.init();

        f.pack();
         f.setSize(new Dimension(600, 600));
         f.setVisible(true);
        }

    //Make java method for transforming from x_model_space to x_screen_space

        public static int xm2s_transform(double xm) {
                    int xs=0;
                        double xscale_factor=0;
                        double xoffset=0;

                        //set xmodel_space display_limits
                                int xm_min=-10;
                                int xm_max=10;
                        //set xscreen_space display_limits
                                int xs_min=0;
                                int xs_max=600;

                //define xmodel to xscreen_space xscale_factor
                                xscale_factor= (xs_max- xs_min)/(xm_max-xm_min);
                //define xmodel to xscreen_space xoffset
                                xoffset= -xscale_factor*xm_min;
```

```
                    xs = (int)(xscale_factor*xm + xoffset);
            return xs;
        }


    //Make java method for transforming from y_model_space to y_screen_space


            public static int ym2s_transform(double ym) {
                int ys=0;
                double yscale_factor=0;
                double yoffset=0;
                //set ymodel_space display_limits
                    int ym_min=-10;
                    int ym_max=10;
            //set yscreen_space display_limits
                    int ys_min=600;
                    int ys_max=0;
                //define ymodel to yscreen space yscale_factor
                    yscale_factor= (ys_max- ys_min)/(ym_max-ym_min);
                //define ymodel to yscreen space yoffset
                    yoffset= -yscale_factor*ym_max;
            ys = (int)(yscale_factor*ym + yoffset);
            return ys;
        }
   // Make java method for circle equation
            public static double circle_eqn_offset( int sign, double Radius, double x, double xoffset,
double yoffset) {
```

```
                        double y=0;
                        y= yoffset + sign*Math.sqrt(Math.pow(Radius,2) - Math.pow(x-xoffset,2));
                        return y;


                }
    // Make java method for face_object
        public static void face_object( Graphics g, double Radius, int half, double xoffset, double yoffset)
{
        Graphics2D g2 = (Graphics2D) g;
                //g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
                g2.setPaint(Color.gray);



                int sign=1;
        int circle_pts=400;
                double xmod=0;
        double ymod=0;
        int xs=0;
                int ys=0;
                int lower=1;
                int upper=1;
                double xmin=0;
                double xmax=0;


      //Calculate circle coordinates in model_space then plot them in screen_space
```

```
// make circle


xmin= -Radius+xoffset;
 xmax= Radius+xoffset;
for(int i=1; i<3; i++) {




if(i==1) { sign=1;} //compute upper half and lower half of circle
        else{sign=-1;}


                for(int j=-circle_pts/2;j<circle_pts/2;j++) { //make circle_pts number of dots on circle's circumference
                //position along x-axis
                xmod= (xmin)+(double)(xmax-xmin)*2*(double)j/(double)circle_pts;


            //plot only upper or lower half when requested
                // select which half is visible half=0 neither; ==1 upper half, ==2 lower half
                // half ===3 right half, == 4 left half, >=5 both halfs
                if   (half==0)
                        {continue;}   // plot no points on circle
        else if ((sign==-1)&&(half==1))//ignore bottom half
                        {continue;}
                else if ((sign==1)&&(half==2)) //ignore top half
                        {continue;}
                else if ((xmod<=xoffset)&&(half==3)) //ignore left half
                        {continue;}
```

```
                    else if ((xmod>=xoffset)&&(half==4))  //ignore right half
                        {continue;}
                    else
                        {     } // plot all points on circle

                //Define circle's points on circumference in model space


                    ymod= circle_eqn_offset(sign, Radius, xmod, xoffset,yoffset);

            //System.out.println(" xmod "+ xmod );
                    //System.out.println( " ymod "+ ymod);
            //System.out.println( " sign "+ sign);

                    //Transform circle's point on circumference from model space to screen space
                    xs= xm2s_transform(xmod);
                    ys= ym2s_transform(ymod);
                    //Draw point in Applet window
                 g2.draw(new Line2D.Double(xs, ys, xs, ys));
                     }

                }

        }

}
```

December, 2024

**End of Java Source Code**

# QB64 Example

The second exercise will use QB64, a freeware computer language that imitates the MS DOS QBASIC application that was provided in Windows 95 and 98 except it will run on Windows XP and Windows 7. The BASIC acronym stands form Beginner's All –purpose Symbolic Instruction Code. You can Download Qb64 freeware from www.qb64.com. The skills developed in the QB45 exercises can be quickly translated into Visual Basic and (VBA) language skills with little effort. Visual Basic for Applications (VBA) will enable students to cultivate Office Automation skills where MS Office products such as MSWord and MS Excel are programmed to do work instead of using menus and mouse clicks. This saves time and effort especially when working with database like MS Access.

Create smiling face using the source code below marked by "BEGINNING" and "END" OF SOURCE CODE.

1) Start QB64. You may have to locate the QB64 directory and double-click on the QB64.exe icon within the directory. When QB64 starts up it will appear as shown in Figure1.

2) Now, copy the text as it appears below in to the Integrated Development Environment (IDE) of Qb64. Run the program from the top menu bar. To copy highlight the selected text, the press keyboard buttons <ctrl> and <C> at the same time.

3) Place your cursor in the blank area of QB64 IDE then press <ctrl> and <V> at the same time. See Figure 2 for a display of the screen once the source is loaded into the QB64 IDE

4) Then run the source program below in QB64 IDE. Press Run at top of QB64 IDE. See Figure 3

5) The Smiling Face will appear as shown in Figure 4. Press any key on the keyboard to return to the QB64 IDE.

6) Edit a selected parameter within the program to experiment with the variables' influences upon the displayed image

7) Rerun the program with the edited parameter

8) Repeat Steps 5-7 as necessary

9) See if you can add the missing right eyebrow by adding more source code to the program. Hint look at the left eyebrow source code to get an idea.

See the various figures below to paste the source code in to the QB64 IDE.

December, 2024

*******BEGINNING OF SOURCE CODE************************

```
SCREEN 12 'Set screen resolution to 640 pixels X 480 pixels 16 colors
WINDOW (-10, 10)-(10, -10) 'Make window 10 units high, 10 units wide
LINE (0, 10)-(0, -10) 'make y-axis line in center of screen
LINE (10, 0)-(-10, 0) ' make x- axis line in center of screen
'Sketch a face using equation x^2 + y^2= R^2 with offsets
' Draw head
FOR I = 1 TO 2
   IF I = 1 THEN sign = 1 ELSE sign = -1 'if sign is positive upper half of circle is plotted
   FOR x = -9 TO 9 STEP .1 'this is the independent variable
      y = sign * SQR(9 ^ 2 - (x) ^ 2) ' This command solves for y in the circle eqn
      PSET (480 / 640 * x, y) ' this command creates a point . Scale x variable to be equal with y
   NEXT x
NEXT I
'Draw Mouth
FOR x = -7 TO 7 STEP .1
            y = -SQR(7 ^ 2 - (x) ^ 2)
           PSET (480 / 640 * x, y)
   'SLEEP
NEXT x
'Draw Nose
FOR I = 1 TO 2
   IF I = 1 THEN sign = 1 ELSE sign = -1
   FOR x = -2 TO 2 STEP .1
      y = sign * SQR(2 ^ 2 - (x) ^ 2) ' This command solves for y in the circle eqn
      PSET (480 / 640 * x, y) ' this command creates a point
```

```
      'SLEEP
   NEXT x
NEXT I



'Draw right eye
FOR I = 1 TO 2
   IF I = 1 THEN sign = 1 ELSE sign = -1
   FOR x = 2 TO 4 STEP .1
      y = 3 + sign * SQR(1 ^ 2 - (x - 3) ^ 2) ' This command solves for y in the circle eqn
      PSET (480 / 640 * x, y) ' this command creates a point
   NEXT x
NEXT I
'Draw Left eye
FOR I = 1 TO 2
           IF I = 1 THEN sign = 1 ELSE sign = -1
           FOR x = -4 TO -2 STEP .1
           y = 3 + sign * SQR(1 ^ 2 - (x + 3) ^ 2) ' This command solves for y in the circle eqn
          PSET (480 / 640 * x, y) ' this command creates a point
      'SLEEP
           NEXT x
NEXT I
' Left Eye brow
FOR x = -5 TO -3 STEP .1
   y = 3 + SQR(2 ^ 2 - (x + 3) ^ 2) ' This command solves for y in the circle eqn
```

December, 2024

```
    PSET (480 / 640 * x, y) ' this command creates a point
    'SLEEP
NEXT x
'Right Eye  brow
'Can you make the right eye brow?
LOCATE 30, 1 'move cursor to 30th row, 1st column
PRINT " Press any key to continue" 'write this text to screen

SLEEP ' Pause the program until user presses any key on key board
STOP
```

*******END OF SOURCE CODE***********************

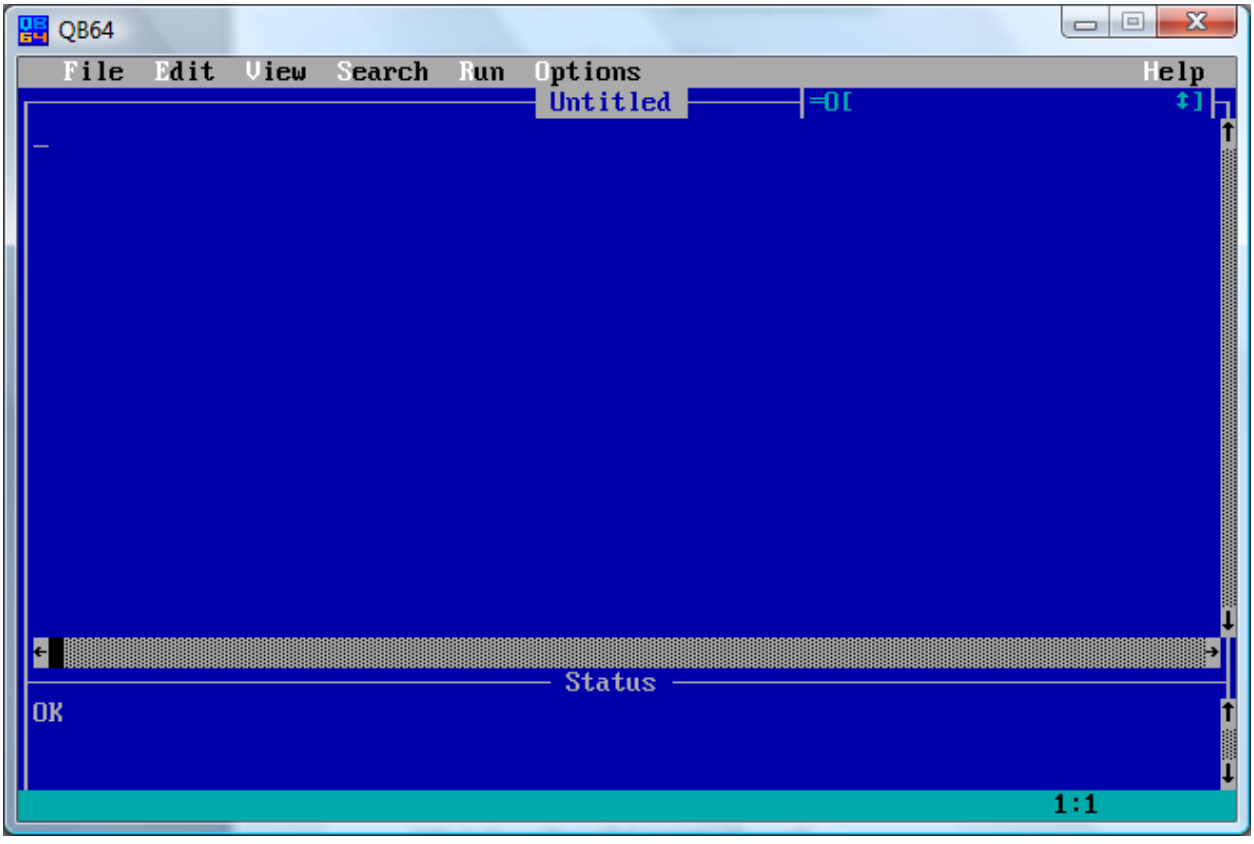



**Figure 2 BLANK QB64 IDE SCREEN AFTER START-UP**

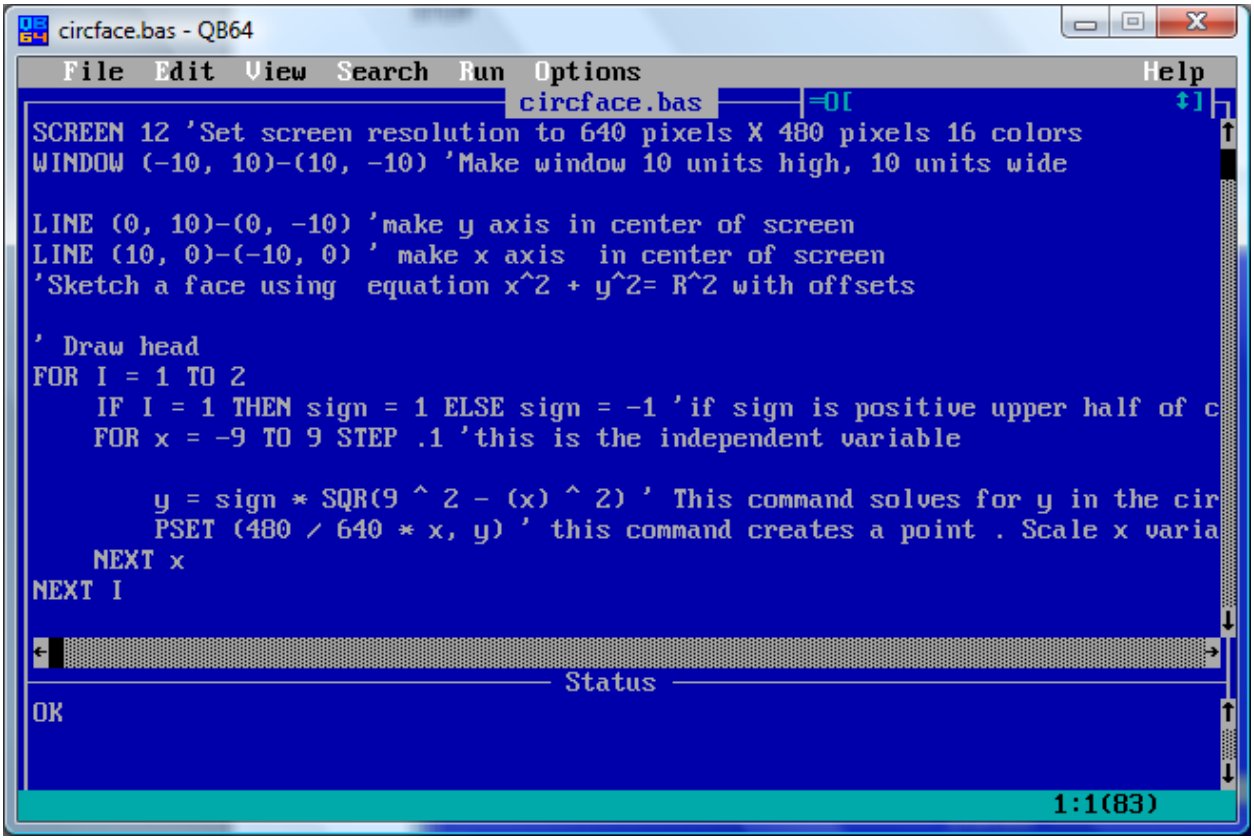



**Figure 3, QB64 IDE AFTER SOURCE CODE IS INSERTED INTO IDE**

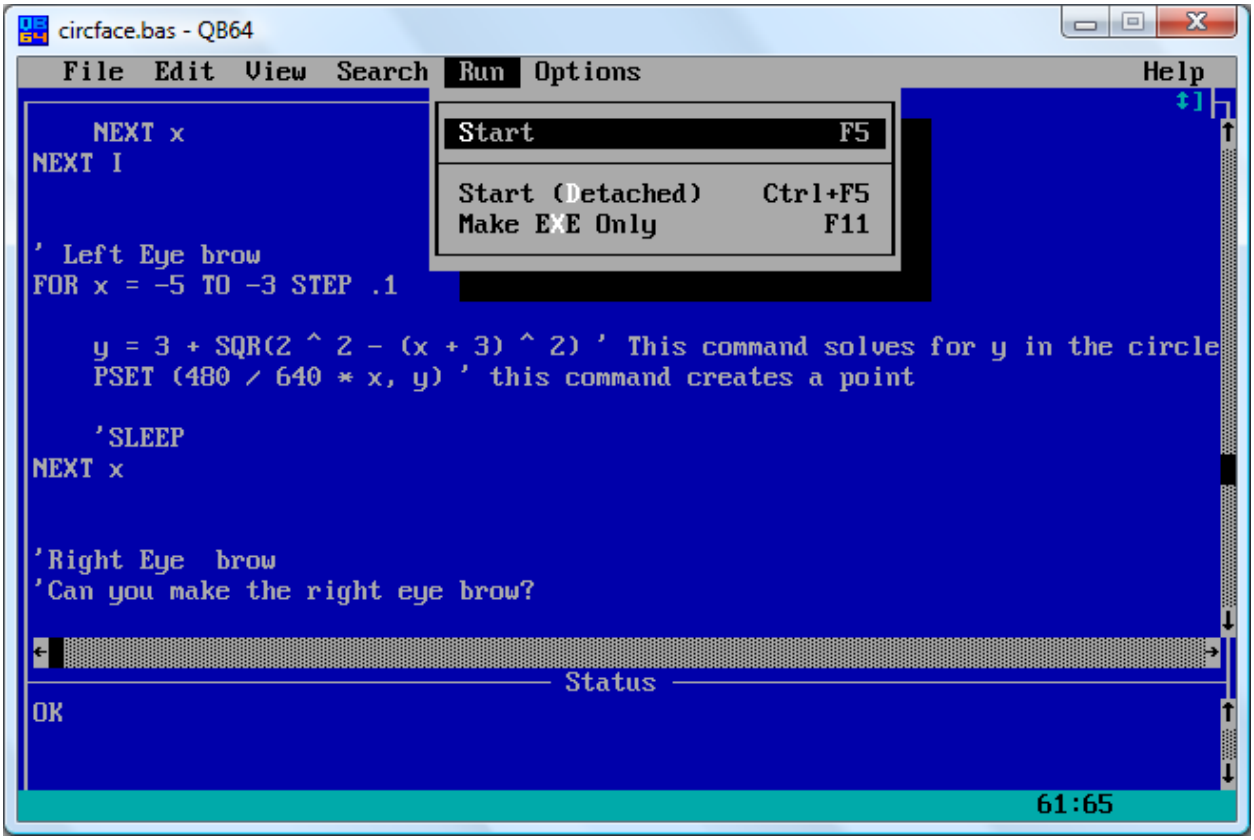



**Figure 4 Starting Program within the QB64 IDE**

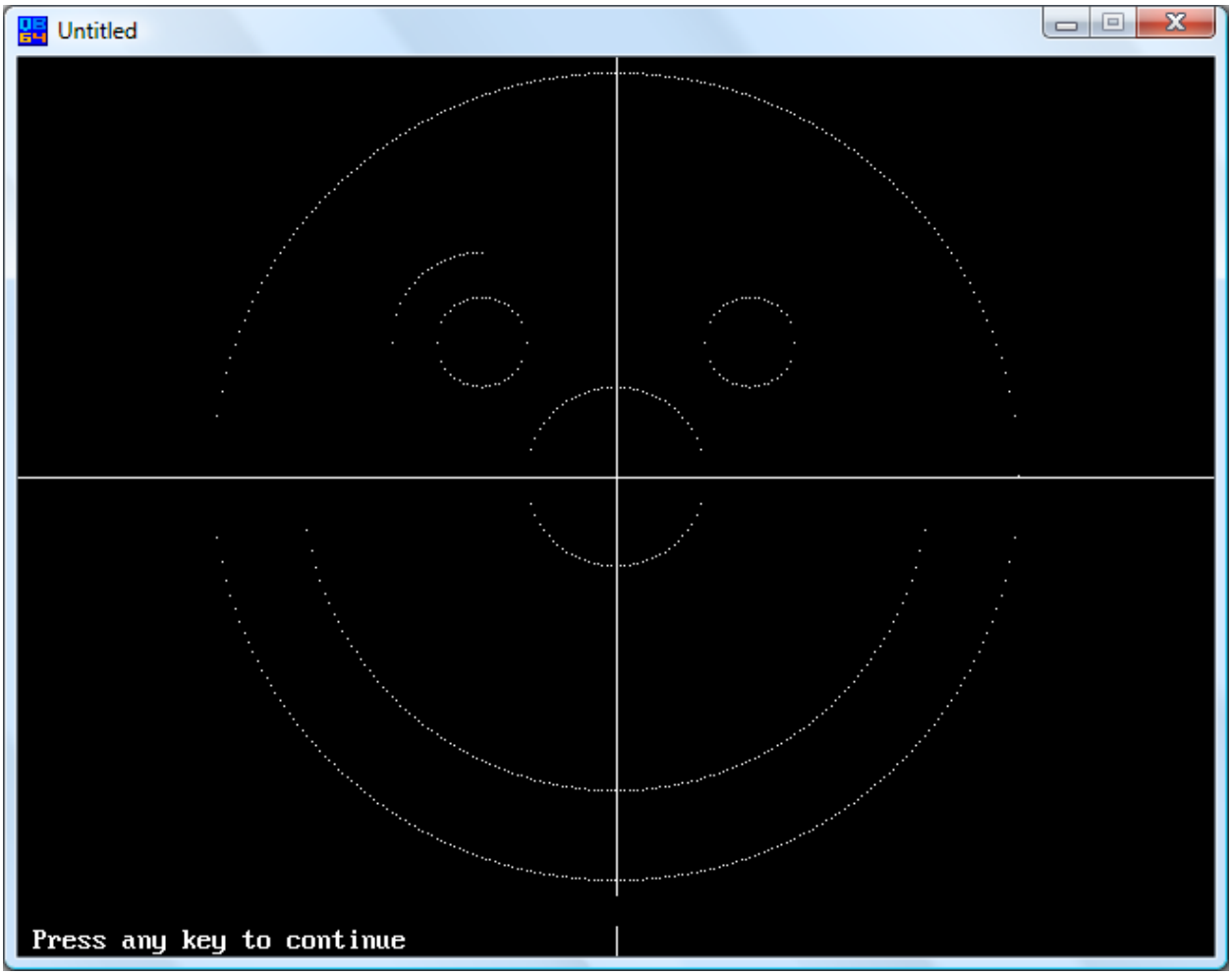



**Figure 5, Smiling Face Screen Display , Second Window appears after Source Code is pasted into QB64 IDE and is Started**