Name: Md. Danial Islam
Id: 20101534
Section : 03

File Name: argtaker.py

```
import argparse
parser = argparse.ArgumentParser()
parser.add_argument("--task", "-t", help="Task number to run", type=int)
parser.add_argument("--stdId", "-id", help="Student Id to show", type=str)
args = parser.parse_args()
```

File Name: windowsize.py

```
# window size, and window position on screen
windowX,windowY = 500,600
windowPosX,windowPosY = 700,300
windowName = "CSE423 - Lab 1 - Md. Danial Islam"
```

File Name: helper.py

```
import random
from argtaker import args
from windowsize import *
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *
def get_random_color():
    return random.random(), random.random(), random.random()
def get_random_point(start,end):
    return random.randint(start,end)
def getTaskNumber():
    if(args.task):
        return args.task
    else:
        return int(input("Please enter task number: "))
def draw_line(x1, y1,x2,y2,pixel_size=5):
    glBegin(GL_LINES)
    glVertex2f(x1,y1)
    glVertex2f(x2,y2)
    glEnd()
def draw_point(x, y,pixel_size=5):
    glPointSize(pixel_size)
    glBegin(GL_POINTS)
    glVertex2f(x,y)
    glEnd()
def draw_triangle_hollow(x1, y1,x2,y2,x3,y3,pixel_size=5):
    glBegin(GL_LINES)
    glVertex2f(x1,y1)
    glVertex2f(x2,y2)
    glVertex2f(x2,y2)
    glVertex2f(x3,y3)
    glVertex2f(x3,y3)
    glVertex2f(x1,y1)
    glEnd()
def draw_quad_hollow(x1, y1,x2,y2,x3,y3,x4,y4,pixel_size=5):
    glBegin(GL_LINES)
    glVertex2f(x1,y1)
    glVertex2f(x2,y2)
    glVertex2f(x2,y2)
    glVertex2f(x3,y3)
    glVertex2f(x3,y3)
    glVertex2f(x4,y4)
    glVertex2f(x4,y4)
    glVertex2f(x1,y1)
    glEnd()
```

File Name: number.py

```
from helper import draw_line
def draw_numbers(num, x):
    numbers = [zero, one, two, three, four, five, six, seven, eight, nine]
    x = numbers[num](x, 250)
    return x + 20
def zero(x, y):
    draw_line(x, y+20, x, y+40)  # 1
    draw_line(x+20, y+40, x, y+40)  # 2
    draw_line(x+20, y+20, x+20, y+40)  # 3
    draw_line(x+20, y, x+20, y+20)  # 4
    draw_line(x, y, x + 20, y)  # 5
    draw_line(x, y, x, y + 20)  # 6
    return x+10
def one(x, y):
    draw_line(x + 20, y, x + 20, y + 40)  # 3
    draw_line(x + 20, y, x + 20, y + 20)  # 4
    return x + 10
def two(x, y):
```

```python
    draw_line(x + 20, y + 40, x, y + 40)  # 2
    draw_line(x + 20, y+20, x + 20, y + 40)  # 3
    draw_line(x, y, x + 20, y)  # 5
    draw_line(x, y, x, y + 20)  # 6
    draw_line(x, y+20, x+20, y+20)  # 7
    return x + 10
def three(x, y):
    draw_line(x + 20, y + 40, x, y + 40)  # 2
    draw_line(x + 20, y + 20, x + 20, y + 40)  # 3
    draw_line(x + 20, y, x + 20, y + 20)  # 4
    draw_line(x, y, x + 20, y)  # 5
    draw_line(x, y + 20, x + 20, y + 20)  # 7
    return x + 10
def four(x, y):
    draw_line(x, y + 20, x, y + 40)  # 1
    draw_line(x + 20, y + 20, x + 20, y + 40)  # 3
    draw_line(x + 20, y, x + 20, y + 20)  # 4
    draw_line(x, y + 20, x + 20, y + 20)  # 7
    return x + 10
def five(x, y):
    draw_line(x, y + 20, x, y + 40)  # 1
    draw_line(x + 20, y + 40, x, y + 40)  # 2
    draw_line(x + 20, y, x + 20, y + 20)  # 4
    draw_line(x, y, x + 20, y)  # 5
    draw_line(x, y + 20, x + 20, y + 20)  # 7
    return x + 10
def six(x, y):
    draw_line(x, y + 20, x, y + 40)  # 1
    draw_line(x + 20, y + 40, x, y + 40)  # 2
    draw_line(x + 20, y, x + 20, y + 20)  # 4
    draw_line(x, y, x + 20, y)  # 5
    draw_line(x, y, x, y + 20)  # 6
    draw_line(x, y + 20, x + 20, y + 20)  # 7
    return x + 10
def seven(x, y):
    draw_line(x + 20, y + 40, x, y + 40)  # 2
    draw_line(x + 20, y + 20, x + 20, y + 40)  # 3
    draw_line(x + 20, y, x + 20, y + 20)  # 4
    return x + 10
def eight(x, y):
    draw_line(x, y + 20, x, y + 40)  # 1
    draw_line(x + 20, y + 40, x, y + 40)  # 2
    draw_line(x + 20, y + 20, x + 20, y + 40)  # 3
    draw_line(x + 20, y, x + 20, y + 20)  # 4
    draw_line(x, y, x + 20, y)  # 5
    draw_line(x, y, x, y + 20)  # 6
    draw_line(x, y + 20, x + 20, y + 20)  # 7
    return x + 10
def nine(x, y):
    draw_line(x, y + 20, x, y + 40)  # 1
    draw_line(x + 20, y + 40, x, y + 40)  # 2
    draw_line(x + 20, y + 20, x + 20, y + 40)  # 3
    draw_line(x + 20, y, x + 20, y + 20)  # 4
    draw_line(x, y, x + 20, y)  # 5
    draw_line(x, y + 20, x + 20, y + 20)  # 7
    return x + 10
```

File Name: tasks.py

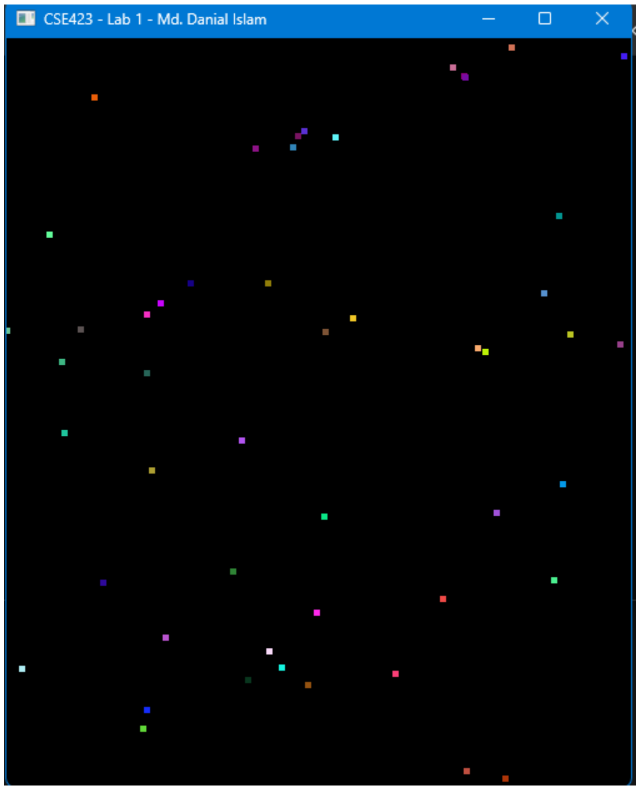
```python
from helper import *
from number import draw_numbers
def task1():
    for i in range(50):
        r,g,b = get_random_color()
        glColor3f(r,g,b)
        randX = get_random_point(0,windowX)
        randY = get_random_point(0,windowY)
        draw_point(randX,randY)
def task2():
    hollowX = 50
    hollowY = 30
    triangleGap = 190
    Ax,Ay = hollowX,hollowY
    Bx,By = windowX-hollowX,Ay
    Cx,Cy = windowX-hollowX,windowY-hollowY-triangleGap
    Dx,Dy = Ax,windowY-hollowY-triangleGap
    midpointX = (windowX-hollowX*2)//2
    Mx,My = hollowX+midpointX,windowY-hollowY
    draw_line( Ax,Ay,  Bx,By )
    draw_line( Ax,Ay,  Dx,Dy)
    draw_line( Bx,By,  Cx,Cy )
    draw_triangle_hollow( Mx,My,   Dx,Dy,   Cx,Cy)
    gapXSide = 20
    gapYSide = 40
    lengthSide = 100
    lwindowUx,lwindowUy = Dx+gapXSide, Dy-gapYSide
    draw_quad_hollow( lwindowUx,lwindowUy,   lwindowUx+lengthSide,lwindowUy,
lwindowUx+lengthSide,lwindowUy-lengthSide,   lwindowUx,lwindowUy-lengthSide )

    RwindowUx,RwindowUy = Cx-gapXSide, Cy-gapYSide
    draw_quad_hollow( RwindowUx,RwindowUy,   RwindowUx-lengthSide,RwindowUy,
RwindowUx-lengthSide,RwindowUy-lengthSide,   RwindowUx,RwindowUy-lengthSide )
    doorGap = lengthSide//2.5
    doorLength = lengthSide*1.3
    draw_quad_hollow( Mx-doorGap,Ay,   Mx-doorGap,Ay+doorLength,
Mx+doorGap,Ay+doorLength, Mx+doorGap,Ay)
    draw_line(Mx-doorGap,Ay+doorLength, Mx*1.09, Ay+doorLength*.9)
    draw_line(Mx*1.09, Ay+doorLength*.9, Mx*1.09, Ay+doorLength*.1)
    draw_line(Mx*1.09, Ay+doorLength*.1, Mx-doorGap,Ay)
    draw_point(Mx*1.06, Ay+doorLength*.5)
def task3():
    x = 130
    if(args.stdId):
        stdId = args.stdId
    else:
        stdId = "20101534"
    for i in stdId:
        r,g,b = get_random_color()
        glColor3f(r,g,b)
        x = draw_numbers(int(i), x)
    glutSwapBuffers()
```

File Name: main.py

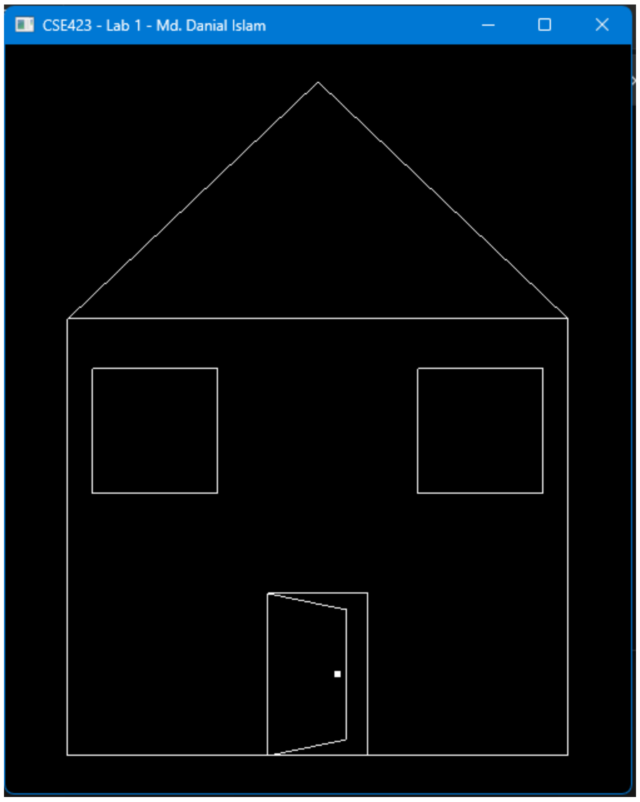
```
from helper import *
from tasks import *
def iterate():
    glViewport(0, 0, windowX, windowY)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    glOrtho(0.0, windowX, 0.0, windowY, 0.0, 1.0)
    glMatrixMode (GL_MODELVIEW)
    glLoadIdentity()
def showScreen(n):
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    iterate()
    task = [task1,task2,task3]
    task[n-1]()
    glutSwapBuffers()

n = getTaskNumber()
glutInit()
glutInitDisplayMode(GLUT_RGBA)
glutInitWindowSize(windowX, windowY)
glutInitWindowPosition(windowPosX, windowPosY)
wind= glutCreateWindow(bytes(windowName, "utf-8"))
glutDisplayFunc(lambda: showScreen(n))
glutMainLoop()
```

Screenshots of the running program

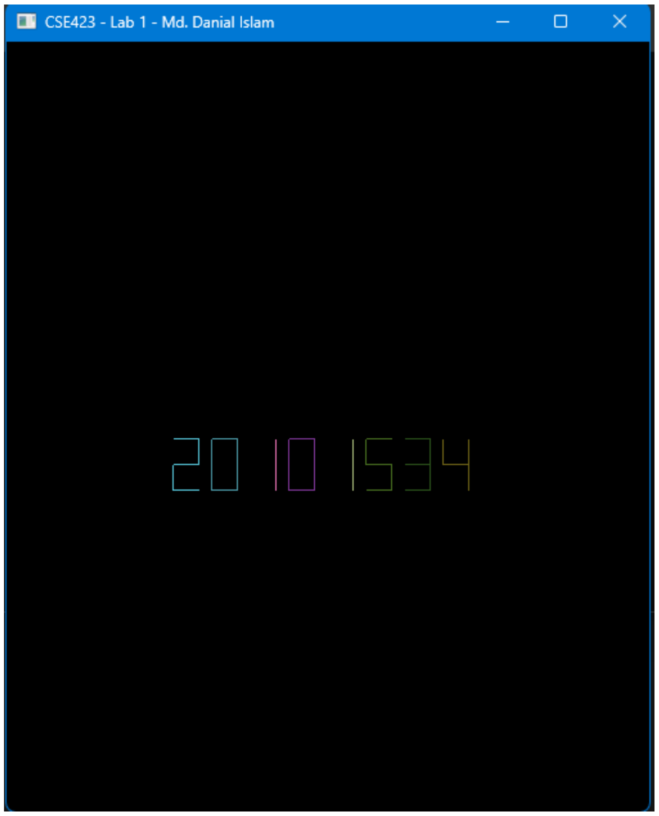
Task 1:



Task 2:



Task 3:


CSE423 - Lab 1 - Md. Danial Islam
20 10 1534