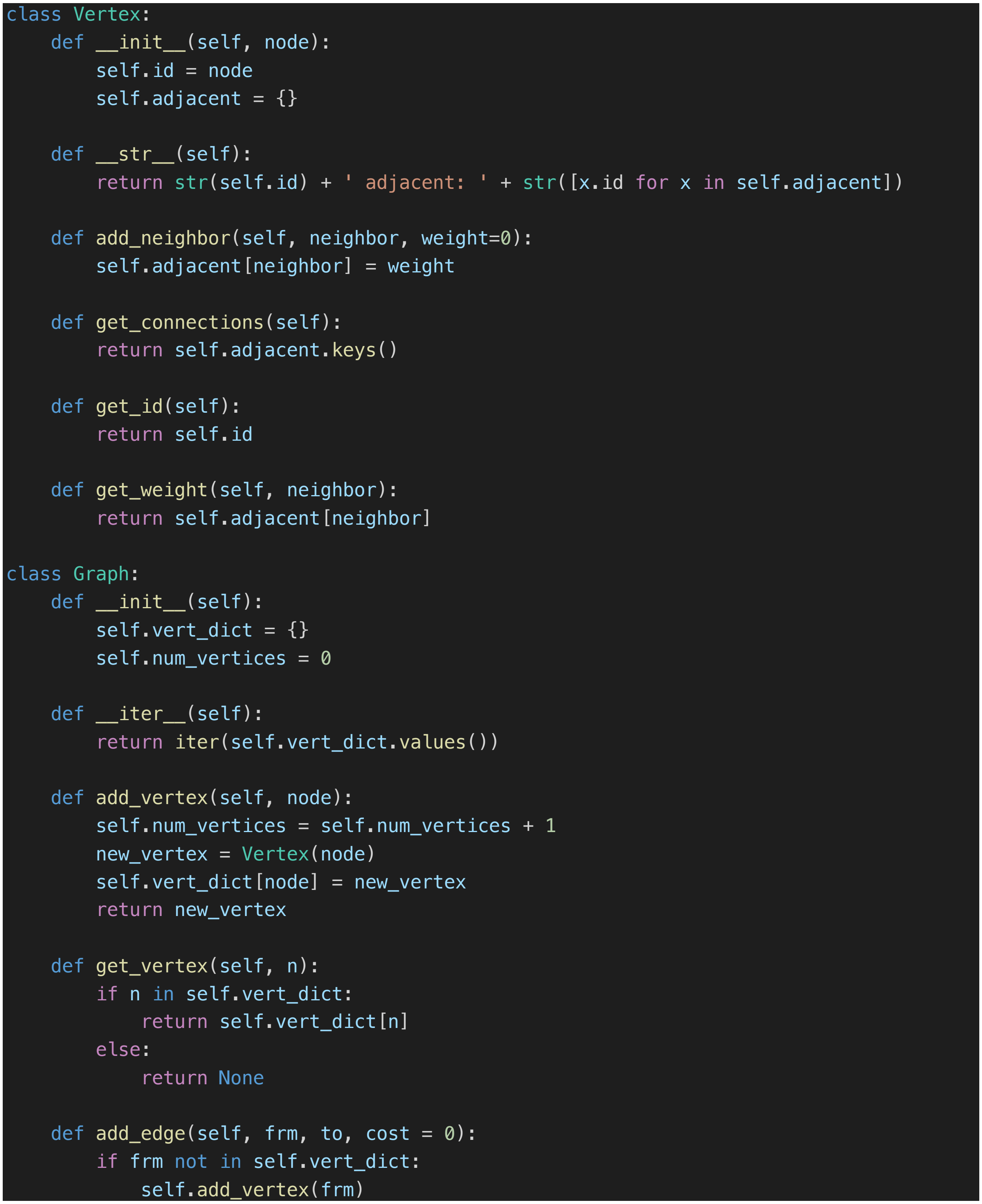
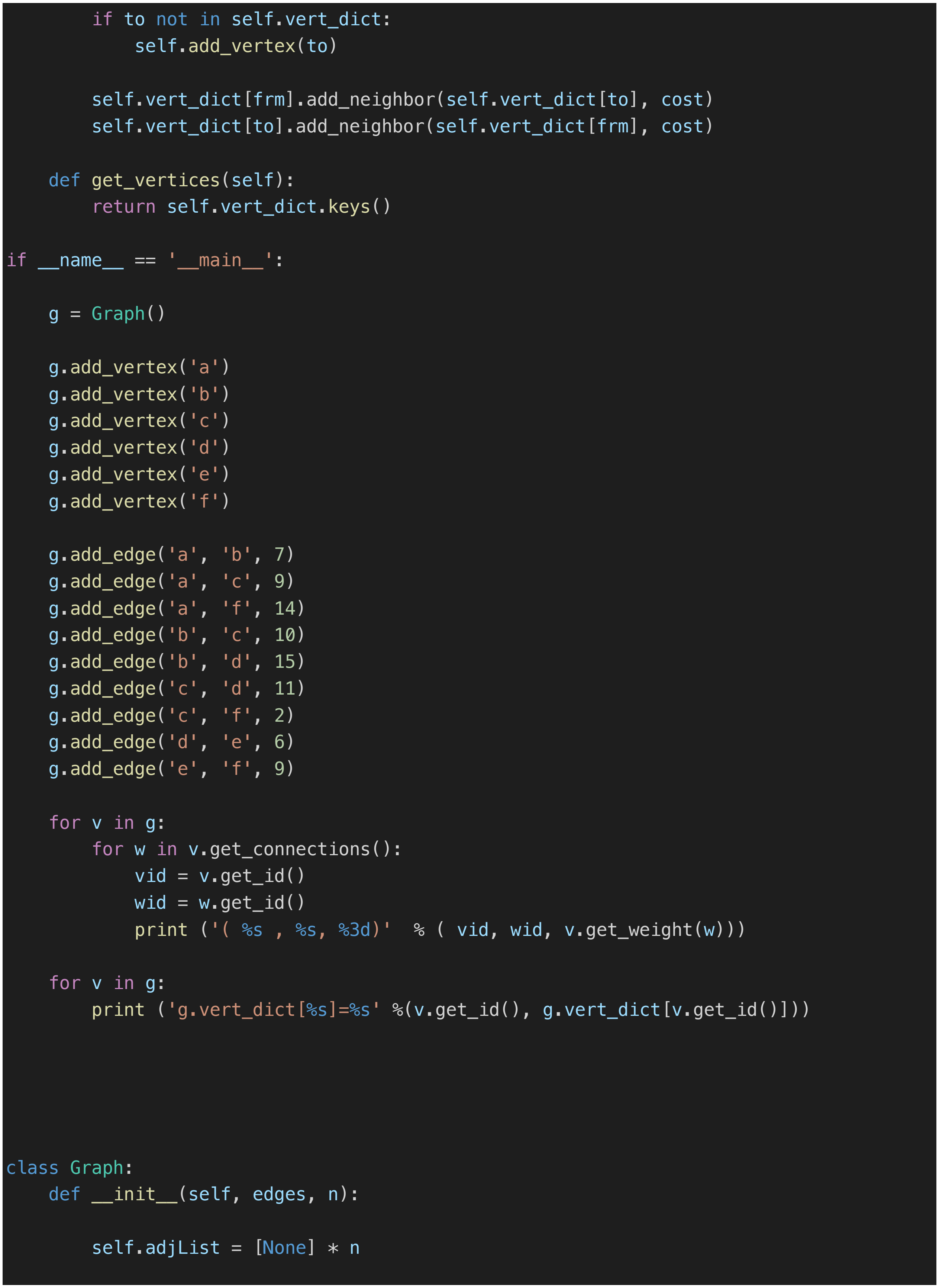
Shriharsha A Vaidhyam

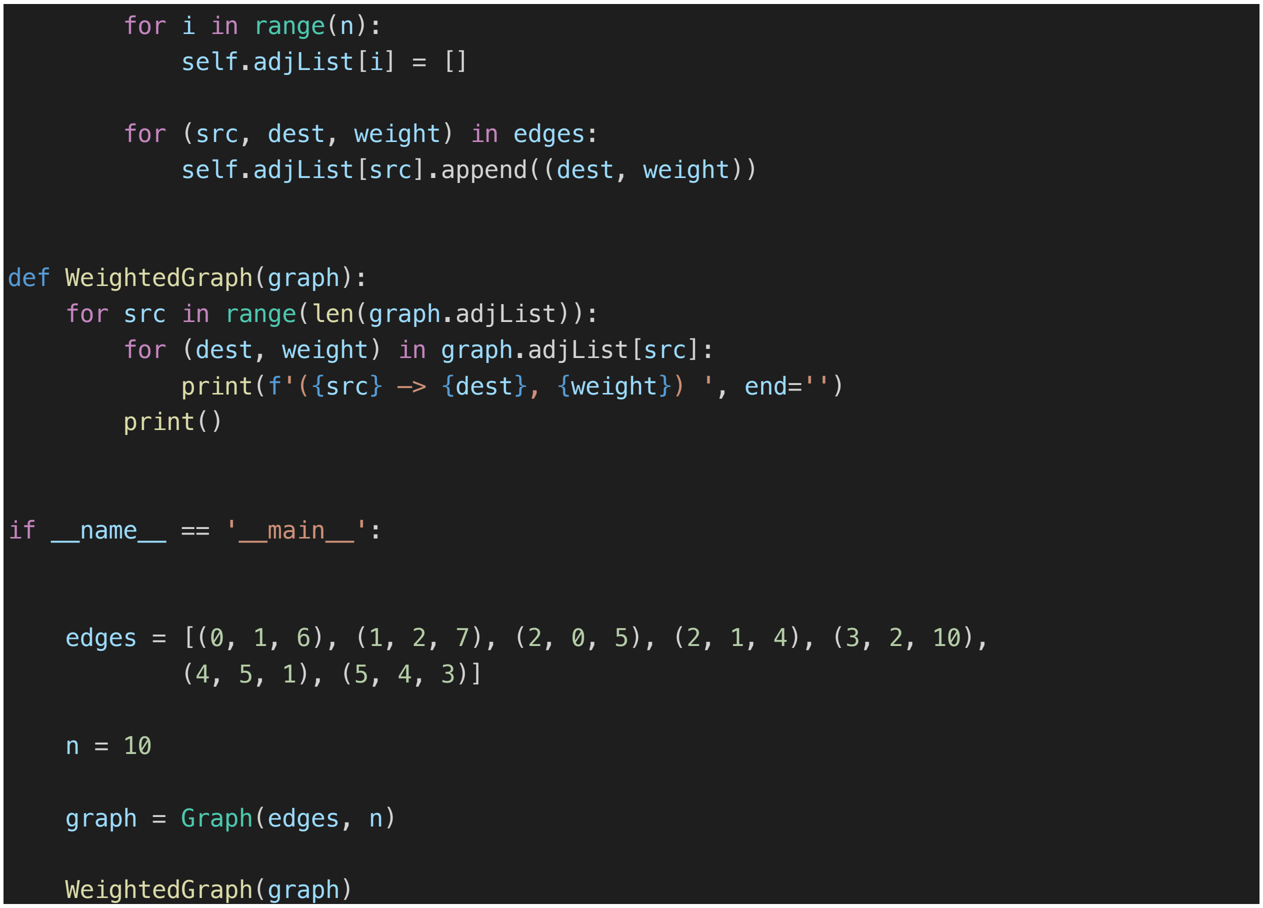
Assignment 4:



```
class Vertex:
    def __init__(self, node):
        self.id = node
        self.adjacent = {}

    def __str__(self):
        return str(self.id) + ' adjacent: ' + str([x.id for x in self.adjacent])

    def add_neighbor(self, neighbor, weight=0):
        self.adjacent[neighbor] = weight

    def get_connections(self):
        return self.adjacent.keys()

    def get_id(self):
        return self.id

    def get_weight(self, neighbor):
        return self.adjacent[neighbor]

class Graph:
    def __init__(self):
        self.vert_dict = {}
        self.num_vertices = 0

    def __iter__(self):
        return iter(self.vert_dict.values())

    def add_vertex(self, node):
        self.num_vertices = self.num_vertices + 1
        new_vertex = Vertex(node)
        self.vert_dict[node] = new_vertex
        return new_vertex

    def get_vertex(self, n):
        if n in self.vert_dict:
            return self.vert_dict[n]
        else:
            return None

    def add_edge(self, frm, to, cost = 0):
        if frm not in self.vert_dict:
            self.add_vertex(frm)
```



```
        if to not in self.vert_dict:
            self.add_vertex(to)

        self.vert_dict[frm].add_neighbor(self.vert_dict[to], cost)
        self.vert_dict[to].add_neighbor(self.vert_dict[frm], cost)

    def get_vertices(self):
        return self.vert_dict.keys()

if __name__ == '__main__':

    g = Graph()

    g.add_vertex('a')
    g.add_vertex('b')
    g.add_vertex('c')
    g.add_vertex('d')
    g.add_vertex('e')
    g.add_vertex('f')

    g.add_edge('a', 'b', 7)
    g.add_edge('a', 'c', 9)
    g.add_edge('a', 'f', 14)
    g.add_edge('b', 'c', 10)
    g.add_edge('b', 'd', 15)
    g.add_edge('c', 'd', 11)
    g.add_edge('c', 'f', 2)
    g.add_edge('d', 'e', 6)
    g.add_edge('e', 'f', 9)

    for v in g:
        for w in v.get_connections():
            vid = v.get_id()
            wid = w.get_id()
            print ('( %s , %s, %3d)'  % ( vid, wid, v.get_weight(w)))

    for v in g:
        print ('g.vert_dict[%s]=%s' %(v.get_id(), g.vert_dict[v.get_id()]))




class Graph:
    def __init__(self, edges, n):

        self.adjList = [None] * n
```



```
        for i in range(n):
            self.adjList[i] = []

        for (src, dest, weight) in edges:
            self.adjList[src].append((dest, weight))


def WeightedGraph(graph):
    for src in range(len(graph.adjList)):
        for (dest, weight) in graph.adjList[src]:
            print(f'({src} —> {dest}, {weight}) ', end='')
        print()


if __name__ == '__main__':


    edges = [(0, 1, 6), (1, 2, 7), (2, 0, 5), (2, 1, 4), (3, 2, 10),
            (4, 5, 1), (5, 4, 3)]

    n = 10

    graph = Graph(edges, n)

    WeightedGraph(graph)
```

Output:

```
harshavaidhyam@Harshas-MacBook-Pro Algo Design %  cd
/Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ Design ; /usr/bin/env
/usr/local/bin/python3 /Users/harshavaidhyam
/.vscode/extensions/ms-python.python-
2022.14.0/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 64408 --
/Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ D
esign/assignement4.py
( a , b,   7)
( a , c,   9)
( a , f,  14)
( b , a,   7)
( b , c,  10)
( b , d,  15)
( c , a,   9)
( c , b,  10)
( c , d,  11)
( c , f,   2)
( d , b,  15)
( d , c,  11)
( d , e,   6)
( e , d,   6)
( e , f,   9)
( f , a,  14)
( f , c,   2)
( f , e,   9)
g.vert_dict[a]=a adjacent: ['b', 'c', 'f']
g.vert_dict[b]=b adjacent: ['a', 'c', 'd']
g.vert_dict[c]=c adjacent: ['a', 'b', 'd', 'f']
g.vert_dict[d]=d adjacent: ['b', 'c', 'e']
g.vert_dict[e]=e adjacent: ['d', 'f']
g.vert_dict[f]=f adjacent: ['a', 'c', 'e']
harshavaidhyam@Harshas-MacBook-Pro Algo Design %  cd
/Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ Design ; /usr/bin/env
/usr/local/bin/python3 /Users/harshavaidhyam
/.vscode/extensions/ms-python.python-
2022.14.0/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 64910 --
/Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ D
esign/assignement4.py
(0 —> 1, 6)
(1 —> 2, 7)
(2 —> 0, 5) (2 —> 1, 4)
(3 —> 2, 10)
(4 —> 5, 1)
(5 —> 4, 3)
harshavaidhyam@Harshas-MacBook-Pro Algo Design %  cd
/Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ Design ; /usr/bin/env
/usr/local/bin/python3 /Users/harshavaidhyam
/.vscode/extensions/ms-python.python-
2022.14.0/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 64921 --
/Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ D
esign/assignement4.py
(0 —> 1, 6)
(1 —> 2, 7)
(2 —> 0, 5) (2 —> 1, 4)
(3 —> 2, 10)
(4 —> 5, 1)
(5 —> 4, 3)
```

```
harshavaidhyam@Harshas-MacBook-Pro Algo Design %  cd
/Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ Design ; /usr/bin/env
/usr/local/bin/python3 /Users/harshavaidhyam
/.vscode/extensions/ms-python.python-
2022.14.0/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 64929 --
/Users/harshavaidhyam/Desktop/Pitt\ term-1/Algo\ D
esign/assignement4.py
(0 —> 1, 6)
(1 —> 2, 7)
(2 —> 0, 5) (2 —> 1, 4)
(3 —> 2, 10)
(4 —> 5, 1)
(5 —> 4, 3)


( a , b,   7)
( a , c,   9)
( a , f,  14)
( b , a,   7)
( b , c,  10)
( b , d,  15)
( c , a,   9)
( c , b,  10)
( c , d,  11)
( c , f,   2)
( d , b,  15)
( d , c,  11)
( d , e,   6)
( e , d,   6)
( e , f,   9)
( f , a,  14)
( f , c,   2)
( f , e,   9)
```