Name: ______________________________ Andrew Id: ______________________

15-112 Spring 2022 Quiz 6

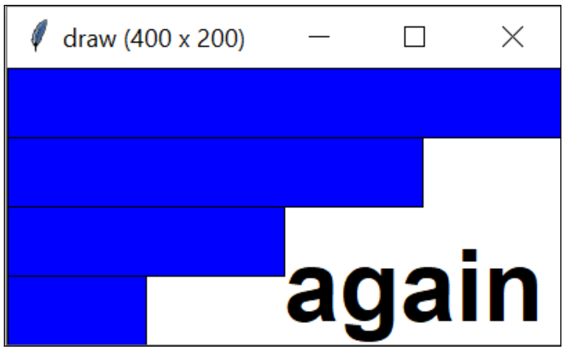
Up to 20 minutes. No calculators, no notes, no books, no computers. Show your work!
Do not use dictionaries, sets, try/except, or recursion on this quiz.

1. (8 points) **Code Tracing**: Indicate what the following program prints. Place your answer (and nothing else) in the box next to the code.

```python
class A(object):
    def __init__(self, x, y):
        self.x = x
        self.y = y
        print("A")

    def __repr__(self):
        return f"A({self.x}, {self.y})"

    def bar(self):
        return self.x + self.y

    def __eq__(self, other):
        return self.y == other.y

class B(A):
    def __init__(self, x):
        super().__init__(3, x)

    def __repr__(self):
        return f"B({self.x})"

class C(A):
    def __init__(self, x):
        super().__init__(x, 3)
        print("C")

    def bar(self):
        return self.x*self.y

a1 = A(1, 3)
a2 = A(0, 2)
b= B(2)
c = C(1)

print(a1 == a2)
print(a1 == b)
print(a1 == c)

print(type(a1) == type(b))
print(type(a1) == type(c))

print(isinstance(a2, type(a1)))
print(isinstance(b, type(a1)))
print(isinstance(c, type(a1)))

print(a1, a1.bar())
print(a2, a2.bar())
print(b, b.bar())
print(c, c.bar())
```

2. (4 points) **Fill-in-the-blanks:** Graphics

Fill in the blanks (A,B,C,D) so that this code draws the given image (shown below). Do not add any new lines. Just fill in the blanks. The image does not need to resize and you may assume the canvas is 400x200.



```
def draw(canvas, width=400, height=200):
    dx = 100 # Width of first rectangle
    dy = 50 # Height of first rectangle
    for i in range(4):
        x0 = 0

        y0 = _________________ # (A)

        x1 = _________________ # (B)

        y1 = __________________# (C)
        canvas.create_rectangle(x0, y0, x1, y1, fill="blue")

    canvas.create_text(200, 200, text='again',

                       anchor=________, # (D)
                       font='Arial 36 bold')

basic_graphics.run(width=400, height=200)
```

3. (10 points) **Free Response: Tasks, TaskList, PrioritizedTaskList**

   Write the classes `Task`, `TaskList`, and `PrioritizedTaskList` so that the following test code runs without errors. Do not hardcode against the values used in the testcases, though you can assume the testcases cover the needed functionality. You must use proper object-oriented design, including good inheritance, or you will lose points!

```
# Tasks have a description, and a priority.
hw = Task("study 112", 4)
clean = Task("clean room", 3)

# You can compare tasks. Tasks are equal if they have the
# same description (case insensitive) and priority.
assert(hw != clean)
assert(hw == Task("study 112", 4))
assert(hw == Task("StUdY 112", 4))
assert(hw != Task("study 112", 5))

# Tasks can be converted to strings
assert(str(hw) == "Task: study 112, Priority: 4")
assert(str(clean) == "Task: clean room, Priority: 3")

# You can create a task list. Task lists always start empty.
sundayList = TaskList()

# You can add task objects to task lists
sundayList.addTask(clean)
sundayList.addTask(hw)

# Getting the next task gives you the first task you entered
# and then removes that task from the task list.
assert(sundayList.getNextTask() == clean)
assert(sundayList.getNextTask() == hw)

# You can make a prioritized task list too!
netflix = Task("Netflix", 1)
betterSundayList = PrioritizedTaskList()
betterSundayList.addTask(netflix)
betterSundayList.addTask(clean)
betterSundayList.addTask(hw)

# Getting the next task gives you the task with the highest priority
# and then removes that task from the task list.
# Hint: try to split this task into two: find the highest priority first, and
# the remove it from the list
assert(betterSundayList.getNextTask() == hw)
assert(betterSundayList.getNextTask() == clean)
assert(betterSundayList.getNextTask() == netflix)

# Note: These test cases don't consider ties, you may ignore that case.
# Note: These test cases don't consider what happens if the list is empty
# when getNextTask is called, you may ignore that case.
```

Answer space for Question 3