This exam is to be done without any materials, electronic or otherwise, except for one \(8\frac{1}{2}\) × 11 paper sheet of notes.

There are 8 parts for a total of 96 points. The last two are 20 and 30 point programming problems. Answer them all on the question sheets. Incomprehensible answers get zero points! Inefficient or poorly expressed code, and answers that do not display understanding of course content might not earn full or even any credit, even if logically right.

Problem 1: (5 points) Write out the Java “Hello World” program. Use Python-like indentation even though indentation doesn’t matter in Java except for human readability.

Problem 2: (6 points) A type consists of a ______ and a set of ______. Give two examples:

<table>
<thead>
<tr>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>______</td>
<td>Set of ______</td>
</tr>
<tr>
<td>______</td>
<td>Set of ______</td>
</tr>
</tbody>
</table>

Problem 3: (5 points) Define \texttt{main()} in Python so that when \texttt{main()} is called, it does the following, and only the following, in the given order:

1. Print Hi
2. Call function named \texttt{fun1} with two arguments, 3 and \texttt{‘Hasta’}, and then print the return value.
3. Print Bye

Problem 4: (5 points) What’s printed?

```
md = dict(); md[‘a’]=’b’; md[‘b’]=’c’
print md[‘a’]
print md[‘b’]
print md[‘c’]
```
Problem 5: (14 points) Review the code quoted from Kumar, Chapter 7. Observe how lists are used for THREE different purposes. Note that one of those purposes is to enable the `arbitrate()` function to return two numbers at once so those numbers can be used by `main()` to change the left and right robot motor power levels.

Then answer the questions below and on the next page about the other TWO purposes.

```python
def cruise():
    # is always ON, just move forward
    return [True, cruiseSpeed, 0]

def avoid():
    # see if there are any obstacles
    L = getIR('left'); R = getIR('right')  # I simplified this.
    L = 1 - L
    R = 1 - R
    if L:
        return [True, 0, -turnSpeed]
    elif R:
        return [True, 0, turnSpeed]
    else:
        return [False, 0, 0]

# list of behaviors, ordered by priority (left is highest)
behaviors = [seekLight, avoid, cruise]
def main():
    while True:
        T, R = arbitrate()
        move(T, R)  # non-stopping robot motor set command.

main()

# Decide which behavior, in order of priority
# has a recommendation for the robot

def arbitrate():
    for behavior in behaviors:
        output, T, R = behavior()
        if output:
            return [T, R]  # constructs list for motor power setting purpose.
```

1. (2 points) The last line constructs a list and returns it. Circle all 5 of the OTHER expressions that construct a list and mark them each with 1.

Problem continued on the next page.
2. (6 points) (a) In one of the ways lists are used, the list has (how many) ___ entries.
   (b) Mark with a 2 all the places where a list used THAT way is constructed or otherwise accessed or used.
   (c) Is it correct to change the number of entries in THOSE lists without changing the rest of the program (except for adding or removing functions like \texttt{cruise()} or \texttt{avoid()}).? Write YES or NO:______.
   (d) Now explain what EACH of the list entries in one of THOSE lists is used for.

3. (6 points) (a) In the OTHER one of the ways lists are used, the list has (how many) ___ entries.
   (b) Mark with a 3 all the places where a list used that OTHER way is constructed or otherwise accessed or used.
   (c) Is it correct to change the number of entries in those OTHER lists without changing the rest of the program (except for adding or removing functions like \texttt{cruise()} or \texttt{avoid()}).? Write YES or NO:______.
   (d) Now explain what EACH of the list entries in one of those OTHER lists is used for.
Problem 6: (16 points) Suppose $a$ and $b$ are integers and $b \gg a$ ($b$ is much greater than $a$). The integral$^1$ of a function $g$ over the interval $[a, b] = \{x \text{ for which } x \geq a \text{ and } x \leq b\}$ can be approximated by adding the function values at $x = a, a + 1, \ldots b - 1$. (Note the last! Also, don’t confuse this use of mathematics notation for closed interval with Python syntax for list construction and input/output.)

1. (2 points) How many integers are in the interval $[a, b]$? ________

2. (2 points) How many function values should be summed together? ________

3. (3 points) Write out the numbers to be added when $g(x) = 4 + x^2$ and the interval is $[3, 6]$. Hint: $3^2 = 9$, $4^2 = 16$, $5^2 = 25$. ________________________________

4. (5 points) Write a Python function $aPIntPr( f, a, b )$ that when given a Python function $f$ and two ints as arguments, PRINTS this approximation to the integral.

5. (4 points) Show what you would type into the IDLE shell window to use it to print the approximate integral of the above function $g$ over the interval $[100, 200]$.

---

$^1$Knowledge of calculus NOT required.
Problem 7: (20 points) Let’s say a pixel is “reddish” if its $R$ intensity both exceeds its $G$ intensity and exceeds its $B$ intensity, i.e., $R > G$ and $R > B$. Similarly, a pixel is “greenish” if its $G$ intensity exceeds both the other two; and a pixel is “bluish” if its $B$ intensity exceeds both the other two.

Write a Python/Myro function that takes a Myro picture as its argument and PRINTS:

(a) How many pixels are reddish. (b) How many are greenish. (c) How many are bluish. (d) How many are neither reddish, greenish nor bluish. (d) Four fractions (printed as floats): proportions of the reddish, greenish, bluish, and remaining pixels. (There can be remaining pixels because two or three of R, G and B intensities can equal one another.)

```python
#Printing statements. You write code for the numbers to print.
print 'Number of reddish pixels:',
print 'Number of greenish pixels:',
print 'Number of bluish pixels:',
print 'Number of other pixels:',
print 'Proportion of reddish pixels:',
print 'Proportion of greenish pixels:',
print 'Proportion of bluish pixels:',
print 'Proportion of other pixels:',
return #no return value
#end of function
```
Problem 8: (30 points) A professor gives a class a 10 point quiz and wants a Python program that prints a histogram of the grades and some statistics. The grades are integers between 0 and 10 inclusively. The program should behave as follows:

1. (3 points) Prompt for and input the number of students who took the quiz.

2. (5 points) Prompt for and input each grade. (It should work on computing the histogram and perhaps other statistics in between the input of each grade. Remember that the histogram reports the number of students with each grade.)

3. (5 points) Print the histogram.

4. (5 points) Calculate and print the average grade.

(5 points) Consider grades 0, 7, 2, 6, 2, 6, 3, 1, 9, 1, 1, 10, 5; there are 13 grades from 13 students. (a) Draw the histogram by hand. (b) Find by hand an approximation to the median. This is a grade $M$ for which about half the students got $\leq M$ and about half got $\geq M$. (An approximation is requested so you can avoid the complications that are due to many students having the same grade. Hint: When you know the number of students, you can find the approximation to the median by scanning the histogram, from grade 0 upward.)

(7 points) Now write the code that when added to your program will make it calculate the approximate median from the histogram and print it. Hint: Use a `while`. 

\[
\begin{array}{cccccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
| & | & | & | & | & | & | & | & |
| 1 |
| 2 |
| 3 |
| 4 |
\end{array}
\]