Name (print): ________________________________________________

• INSTRUCTIONS:

  – Keep your eyes on your own paper and do your best to prevent anyone else from seeing your work.
  – Do NOT communicate with anyone other than the professor/proctor for ANY reason in ANY language in ANY manner.
  – This exam is closed notes, closed books, and no calculator.
  – Turn all mobile devices off and put them away now. You cannot have them on your desk.
  – Write neatly and clearly indicate your answers. What I cannot read, I will assume to be incorrect.
  – Stop writing when told to do so at the end of the exam. I will take 5 points off your exam if I have to tell you multiple times.
  – Academic misconduct will not be tolerated. Suspected academic misconduct will be immediately referred to the Emory Honor Council. Penalties for misconduct will be a zero on this exam, an F grade in the course, and/or other disciplinary action that may be applied by the Emory Honor Council.

• TIME: This exam has 5 questions on 8 pages including the title page. Please check to make sure all pages are included. You will have 50 minutes to complete this exam.

I commit to uphold the ideals of honor and integrity by refusing to betray the trust bestowed upon me as a member of the Emory community. I have also read and understand the requirements and policies outlined above.

Signature: ________________________________________________

<table>
<thead>
<tr>
<th>Question</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points:</td>
<td>12</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>Score:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. **Short Answer.** Briefly (no more than 2-3 sentences, maximum), answer the following questions:

(a) (3 points) List 3 distinct differences between the Mars rover Curiosity and the Mars rovers Spirit/Opportunity.

**Solution:**
- Power source (nuclear vs. solar)
- Landing mechanism (airbags vs. skycrane)
- Computing power
- Size and weight differences
- Curiosity has more autonomous control (software more advanced)

(b) (2 points) Why was the Unimate robot significant in the history of robotics?

**Solution:**
- First industrial robot. Also first widely successful robot used in factory mechanization.

(c) (2 points) Python (and more generally, programming languages) are said to be formal languages. Why must computers use formal languages rather than natural languages (like humans speak)?

**Solution:** Natural languages are imprecise and computers require exactly 1 interpretation of a command in order to properly execute. (Ch1/p.11)
(d) (3 points) Describe (briefly!) how robots use stereo-vision to “see” terrain using cameras.

**Solution:** Need at least 2 cameras mounted at slightly different angles somewhere on the robot. The robot subtracts these images and looks at the areas of difference to try to identify obstacles.

(e) (2 points) In the short story, “Moxon’s Master,” the word *robot* is never used. Instead, the author uses the term “automaton.” Why didn’t the author use the word *robot*?

**Solution:** The word robot was invented until Karel Capek’s play “R.U.R” in the 1920’s. “Moxon’s Master” was written 30 years before that.
2. (8 points) **Operator Speaking**

Write a function called `mathOps` which takes two integer parameters. This function should use the parameter to print out:

- their sum
- the difference of the first and the second number
- their product
- the quotient of the first and the second number
- the remainder of the first divided by the second number
- the first number raised to the power of the second number

**Solution:** Solutions vary but one possible solution:

```python
def MathOps(x, y):
    print x+y
    print x-y
    print x*y
    print x/y
    print x%y
    print x**y
```

Scoring:
+1 function definition
+6 (1pt each) for mathematical operators/operations
+1 printing
3. (10 points) **LONG TIME PASSING**

Write a function called `secsToTime` which gets an integer (representing a number of seconds) from the user. Your function should print out the number of complete minutes represented by that number of seconds and the “leftover” seconds. Examples of output (with specific user input) when the function is invoked are below.

```python
>>> secsToTime()
How many seconds do you have? 45
45 seconds is 0 minutes 45 seconds
>>> secsToTime()
How many seconds do you have? 60
60 seconds is 1 minutes 0 seconds
>>> secsToTime()
How many seconds do you have? 145
145 seconds is 2 minutes 25 seconds
```

**Solution:** Solutions vary but something like:

```python
def secsToTime():
    s = input("How many seconds do you have? ")
    minutes = s / 60
    remainder = s % 60
    print s, "seconds is", minutes, "minutes", remainder, "seconds"
```

Scoring:

+2 function definition
+2 input from user stored in variable
+2 calculates minutes correctly
+2 calculates “leftover” seconds correctly
+2 prints info correctly
4. **Robot Drawing**

Assume `turn90degrees()` has been defined as below so the robot turns right 90° and `nudge(x)` has been defined to move the robot forward x units.

```python
def turn90degrees():
    turnRight(1, 1)

def nudge(x):
    forward(1, x)
```

The robot’s initial position is indicated by the circle and it always begins facing north/up. The following code makes the robot drive the trajectory drawn in the box to the right.

```
nudge(1)
turn90degrees()
nudge(1)
nudge(2)
```

(a) (6 points) Draw the robot’s trajectory when the following code is executed assuming `turn90degrees()` and `nudge(x)` are defined as above. Label the length of each move (nudge) using numbers as in the example above.

```python
x = 1
y = 2

nudge(x)
x = x + 3
turn90degrees()
nudge(x % 3)
nudge(x/y)
turn90degrees()
nudge(y % x)
turn90degrees()
x = y
# y = x
nudge(4 % (x*y))
```

```
❤
✻
✲
❄
```
(b) (4 points) Write the series of `turn90degrees()` and `nudge()` commands which would lead to the trajectory drawn below.

```
Solution:
nudge(1)
turn90degrees()
nudge(2)
turn90degrees()
nudge(1)
turn90degrees()
turn90degrees()
nudge(1)
turn90degrees()
turn90degrees()
turn90degrees()
nudge(3)
```
5. (10 points) Evaluate each expression. Then give the result of the evaluation and the data type of the result. If the expression cannot be evaluated due to an error, you may simply write “error” for the value. The first row has been done for you.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>4+1</td>
<td>5</td>
<td>int</td>
</tr>
<tr>
<td>4+2.0*5</td>
<td>14.0</td>
<td>float</td>
</tr>
<tr>
<td>3*2/4</td>
<td>1</td>
<td>int</td>
</tr>
<tr>
<td>3-4/8.0</td>
<td>2.5</td>
<td>float</td>
</tr>
<tr>
<td>str(3.0) * 2</td>
<td>“3.03.0”</td>
<td>str</td>
</tr>
<tr>
<td>str(3.0 * 2)</td>
<td>“6.0”</td>
<td>str</td>
</tr>
<tr>
<td>&quot;3&quot; + 6</td>
<td>error</td>
<td></td>
</tr>
<tr>
<td>int(3.0) + 6</td>
<td>9</td>
<td>int</td>
</tr>
<tr>
<td>float(2) + 6</td>
<td>8.0</td>
<td>int</td>
</tr>
<tr>
<td>&quot;CS190&quot; - &quot;190&quot;</td>
<td>error</td>
<td></td>
</tr>
<tr>
<td>&quot;Hello&quot; + &quot; class!&quot;</td>
<td>“Hello class!”</td>
<td>str</td>
</tr>
</tbody>
</table>