

(60-140) Lab Exercises #5

(Due at the end of the lab period or beginning of the next)

Oct. 25-28, 2016.

Objective: Programming with loops

HI-LO game: A random number will be generated. Then, the program prompts for guesses until the user is correct, or has made 10 wrong guesses. After each guess, the program indicates whether the guess was too high, too low, or correct. Once the round has ended, either by a correct guess or by using up the 10 guesses, the program displays the current status. Assuming the random number is 48 (a number between 1 and 50), a sample interaction is shown below

```
Enter your guess (between 1 and 50): 30
Too low ...
Enter your guess (between 1 and 50): 50
Too high ...
Enter your guess (between 1 and 50): 48
Correct, the number was 48.
```

If the user entered more than 10 wrong guesses:

```
Sorry, the number was 48.
```

Part A: RAPTOR Exercise

- Understand more about RAPTOR by watching the video clip about [how to work with loops](#).
- Start RAPTOR and create the flowchart of HI-LO game.
- Save the flowchart to a file named "guessGame.rap" in the working directory on the PC you are using.
- Demonstrate the guessGame.rap file to GA/TAs and run with different input values.

Hint: To generate a random number between 1 and 50 using Raptor, use the following assignment statement

```
goal ← floor((Random*50)+1)
```

For more information, watch the video clip about [how to generate a random number Using Raptor](#)

Part B: C Programming Exercise

- a) Implement the algorithm as represented by “guessGame.rap”, and write an equivalent C program that accomplishes what the flowchart does.
- b) Save your program to a file named “guessGame.c” in the working directory on the PC you are using.
- c) Demonstrate the guessGame.c file to GA/TAs and run with different input values.

Hint: To generate a random number between 1 and 50 in C, first, add `#include <stdlib.h>` to your code and then use the following statement:

```
int goal = rand( ) % 50 + 1;
```

EVALUATION:

You need to show your GA/TA the complete programs at the end of this lab, or at the beginning of your next lab. The marks you will receive for this lab are made of two parts: Lab work marks 10 and attendance marks 5. **Total 15 marks.**

Lab Work Mark: Your C code will be evaluated based on your solutions for the problems based on the following scheme:

1. Does the code run and meet specifications?
 - Is input adequate and input data type properly validated?
 - Is processing adequate?
 - Is output correct and adequate?
 - Is the code compliable? Is the code run properly?
2. Is the code properly commented?
 - Is the program title, programmer's first and last name, and the date posted at the top in a multi-line comment?
 - Is each significant step of the program properly commented?
 - Are comments added to clarify details?
 - Are comments clear, accurate, neatly formatted, and have no misspellings?
3. Is the code properly formatted?
 - Are blocks of code indented according to their parent-child relationship?
 - Do curly braces line up vertically?
 - Is there an empty line between significant steps (blocks) of the program?
 - Is the width of the code contained within a reasonable limit so that minimal horizontal scrolling is required (with 800 x 600 monitor resolution), and there is minimal line-wrapping when printed?
 - Is camel-case notation used for variable, e.g. employeeLastName?

IMPORTANT: DO YOUR OWN CODE. ANY CODE SUSPECTED TO BE SIMILAR TO ANOTHER SUBMISSION WILL CAUSE BOTH SUBMISSIONS TO RECEIVE A ZERO MARK ON ALL LABS AND BE REPORTED FOR PLAGIARISM.