Computer Science 1	**Exercises 08.01-05**	**Date:**
Name:		**Period:**

1. *Program Flow* follows the exact sequence of listed program statements, unless directed other by what?

2. Like *Selection*, *Repetition Control Structures* have a __________ and a certain path that is only followed if it is **True**.

3. What are 2 synonyms for *Repetition*?

4. What loop structure is used with *Fixed Repetition*?

5. In *fixed repetition*, what does *fixed* mean?

6. Compare programs **Repetition01.py** and **Repetition02.py**.
   List 2 reasons why the second program is preferable.

7. What does *LCV* stand for?

8. What is another term for the *LCV*?

Use this statement for questions 9 & 10: `for k in range(5000)`

9. What is the LCV?

10. How many times will this loop repeat?

11. Proper, consistent indentation is necessary for *selection*. Is it also necessary for *repetition*?

12. Look at program **Repetition04.py**. Rewrite line #10 so the program counts up to **40**.

13. Look at program **Repetition06.py**. Rewrite line #11 so the program counts from **1864** to **2091**.

14. Look at program **Repetition07.py**. Rewrite line #12 so the program counts from **33** to **99** by **3**s.

15. Look again at program **Repetition07.py**. Rewrite line #32 so the program counts from **2015** to **1885** backwards by **10**s.

16. Look at program **Repetition08.py**. The program's output is very strange. What causes this?

17. Look at program **Repetition09.py**.
How does this program cure the problem of the previous program?

18. Explain *Condition Repetition*.

19. What command is used for *Conditional Repetition*?

20. How many times will this program display "**HELLO WORLD**" ?

```
for p in range(11):
   for q in range(7):
      print("HELLO WORLD")
```

21. The previous question shows a *Nested Control Structure*, but more specifically, what else does it show?

22. With nested loops, the *inner loop* must complete all of its repetitions __________ the *outer loop* can simply count to the next value.

23. Look again at program **Nested04.py**. Rewrite lines #9-12 so that it will display all times tables from **1** times **1** to **60** times **60** and allow for numbers to be up to 4 digits in length.

24. Compare programs **Nested07.py** and **Nested08.py**. Explain why **Nested08.py** is more practical.

25. Look at program **RepetitionWithGraphics02.py**. Rewrite lines #11-13 so the program will display a regular *octagon*.

26. Look at program **RepetitionWithGraphics03.py**. Right now, the program draws a design comprised of 8 squares. Rewrite lines #11-15 so the program creates a similar design comprised of 12 triangles.