Computer Science 1	**Exercises 04.04-09**	**Date:**
Name:		**Period:**

1. List 2 major differences between variables in computer science and in math.

2. Things like names, words, or even sentences are stored in __________ variables.

3. Look at program **Variables11.py**. Is **zipCode** an *integer* or a *string*? Why?

4. Does Python make a distinction between *strings* and *characters*?

5. In Python, are *string literals* supposed to be placed inside single quotes or double quotes?

6. What is *String Concatenation*?

In questions 7 through 10 print the exact output of the program statement.

7.

```
a = "Tom"
b = "Jones"
c = a + b
print(c)
```

8.

```
a = "Tom"
b = "Jones"
c = a + " " + b
print(c)
```

9.

```
a = "Tom"
b = "Jones"
c = b + ", " + a
print(c)
```

10.

```
a = "66"
b = "23"
c = a + b
print(c)
```

11. Why is the plus sign (+) considered an *Overloaded Operator*?

12. If you add, subtract or multiply 2 integers, the result will be an __________.

13. If you add, subtract or multiply an integer and a real number, the result will be a __________.

14. What data type can only store one of two possible values: **True** and **False**?

15. What does the **type** command do?

16. What is *Type Casting*?

In questions 17 and 18 write the shortcut for the program statement/segment.

17.

```
b = b – 8
```

18.

```
a = 13
b = 13
c = 13
```

In questions 19 through 22 print the exact output of the program statement.

19.

```
n = "Tom "
n += "Jones"
print(n)
```

20.

```
a = 10
b = 20
a = b
b = a
print(a,b)
```

21.

```
a = 10
b = 20
temp = a
a = b
b = temp
print(a,b)
```

22.

```
a = 10
b = 20
a,b = b,a
print(a,b)
```

23. Look at programs **Documentation01.py** and **Documentation02.py**.
Both programs do the same thing. Why is the second program so much easier to understand?

24. Look at program **Documentation03.py** and refer to the previous question.
How does this program further improve readability?

25. When you receive an error message, does it always indicate the correct location of the actual error?

26. Look at program **MoreErrors03.py**. Why does it not execute?

27. Look at program **MoreErrors04.py**. How does this program fix the error of the previous program?

28. Look at program **MoreErrors05.py**. The program has no syntax errors. Why does it still not execute?

29. Look at program **MoreErrors06.py**. The program executes with no error messages whatsoever; however, the wrong average is displayed. Explain the *logic error* and how to fix it.