

I. Answer the following showing the memory simulation for each one: (1.5 pts each)

- 1) What is the output of the following code fragment?
(All variables are of type **int**.)

1) _____

```
number = 1;
for (count = 1; count <= 3; count++)
{
    // optional set of braces
    do
        number = 2 * number;
    while (number <= 5);
}
// optional set of braces
System.out.print(number);
```

MEMORY:
count number

- 2) What is the output of the following code fragment?
(All variables are of type **int**.)

2) _____

```
number = 1;
for (count = 1; count <= 3; count++)
{
    // optional set of braces
    while (number <= 5)
        number = 2 * number;
}
// optional set of braces
System.out.print(number);
```

MEMORY:
count number

- 3) What is the output of the following code fragment with inputs of
5 0 15 -5 2

3) _____

```
sum = 0;
for (count = 1; count <= 5; count++)
{
    String input = JOptionPane.showInputDialog("Type a number");
    number = Integer.parseInt(input);
    if (number <= 0)
        break;
    sum = sum + number;
}
System.out.print(sum);
```

MEMORY:
count number sum

over

4) What is the output of the following code fragment with inputs of
5 0 15 -5 2
sum = 0;
for (count = 1; count <= 5; count++)
{
 String input = JOptionPane.showInputDialog("Type a number");
 number = Integer.parseInt(input);
 if (number <= 0)
 continue;
 sum = sum + number;
}
System.out.print(sum);

4) _____
MEMORY:
count number sum

II. Answer the following

1) Complete the program segment below that will have the following output: (2 pts)

```
for (x = 5; x >= _____ ; _____ )
{
    for (y = 1; y <= _____ ; _____ )
    {
        System.out.print( _____ );
    }
    _____ ;
}
```


**
*

2) Given the code segment below write the output:
(Note: The long memory simulation below is optional.)

(2 pts)

OUTPUT:

```
for (a = 1 ; a <= 2 ; a++)
{
    for (b = 1 ; b <= 3 ; b++)
    {
        for (c = 1 ; c <= 4 ; c++)
        {
            System.out.print("$");
        }
        System.out.println( );
    }
    System.out.println( );
}
```

<u>Memory</u>	<u>Memory continued</u>	<u>Memory continued</u>
<u>a</u> <u>b</u> <u>c</u>	<u>a</u> <u>b</u> <u>c</u>	<u>a</u> <u>b</u> <u>c</u>