Name _____ Period_____

1. Complete the following code segments.

(a) Write code that will assign the value of "Computer Science is Cool" to the String variable "g". Then have it print this String in all lower case.

```
String g = "Computer Science is Cool";
System.out.println(g.toLowerCase());
```

(a) Initialize a String variable b to "Four score and seven years ago". Print all the characters between (not inclusive) the first "s" in score and the "y" in years (EX: "core and seven ")

```
String b = "Four score and seven years ago";
System.out.println(b.substring(7, 22);
```

(c) Write code that will produce the following printout using only a single println(),

Hello
Hello Again

```
System.out.println("Hello\nHello Again");
```

(d) Write code that will print the following to the consol,

"Look here!"

Note, the quotes must display when printed.

```
System.out.println("\"Look here!\"");
```

Name _____ Period_____

2. You need to write a program that generates usernames and passwords for students at Timberline.

(a) The usernames will use the following convention: FirstInitial + LastName (all lower case). The final formatted username should be stored in a variable called username.

Below is sample output that illustrates how your code should run for various users.

firstName	**lastName**	**username**
Lady	Gaga	lgaga
Billie	Eilish	beilish

Write code to generate the properly formatted usernames below,

```
String username = firstName.substring(0, 1) + lastName.substring(0, 1)
username = username.toLowerCase();
```

(b) The passwords must use the following convention: first letter first name + first letter lastname + 6 digit birthdate (all letters lower case). The final formatted username should be stored in a variable called password.

Below is sample output that illustrates how your code should run for various users.

firstName	**lastName**	**bDay**	**password**
Lady	Gaga	03/28/1986	lg032886
Billie	Eilish	12/18/2001	be121801

Write code to generate the properly formatted passwords below,

```
String password = firstName.substring(0, 1).toLowerCase() +
lastName.substring(0, 1).toLowerCase() +
bDay.substring(0, 2) +
bDay.substring(3, 5) +
bDay.substring(bDay.length - 2);
```

3. Write a driver class called WordScramble that extracts the first half and second half of the word, then recombines the halves such that the last half is printed first followed by the first half.

   Below is sample output that illustrates what your code should output for various words,

Word	Output
popcorn	cornpop
bike	kebi
school	oolsch

```
   public class WordScramble{
      public static void main(String args[]){
           System.out.println(word.substring(word.length()/2) +
word.substring(0, word.length()/2);


     }
}
```

4. Write a driver class called “NumericVariables”. In the the driver class, (2 points)

Indicate each part below in your code with appropriate comments. For example, for part a type “//Part a”. (1 point)

1. Declare and initialize an int a1. Write a single line of code that will print to the console the value of a1 decremented by 1. (2 points)
2. Declare and initialize an integer b1. Declare and initialize an integer b2. Then write a single line of code that uses the compound operator, -=, to subtract b2 – 30 from the value of b1 and store the result back in b1. (2 points)
3. Initialize an integer variable c1 and another integer c2 to a value smaller than c1. Write a program that prints the remainder that results when these two numbers are divided. (2 points)
4. Write code that will create a constant variable D1 that is equal 2.718. (1 point)
5. Declare and initialize a double variable e1. Declare and initialize an int variable e2. Declare an int variable e3, and assign the value of e1 divided by e2 to e3. (2 points)

```
public class NumericVariables{

	public static void main(String args[]){
	//part a
	int a1 = 5;
	System.out.println(--a1);

	//part b
	int b1 = 2, int b2 = 3;
	b1 +=(b2 - 30);

	//part c
	int c1 = 5, c2 = 2;
	System.out.println(c1%c2);

	//part d
	final double D1 = 2.718;

	//part e
	double e1 = 5;
	int e2 = 10;
	int e3 = e1/e2;

	}

}
```

/12

5. Write code that could be used to reverse a number and store the reversed number in a variable called `reversed`. Your code should work for any number with 4 digits. Consider the int data type below,

```
int number = 1234;
```

When your code is ran, "4321" should be stored in `reversed`.

Below are more examples,

int data type	**reversed**
`int n1 = 3455;`	5543
`int n2 = 8767;`	7678
`int n3 = 2468;`	8642

```
int n = 1234;
int ones = n%10;
int tens = (n/10)%10;
int hundreds = (n/100)%10;
int thousands = (n/1000)%10;
String reversed = "" + ones + tens + hundreds + thousands;
System.out.println(reversed);
```

/3