Integer.MIN_VALUE → -2147483648
Integer.MAX_VALUE → 2147483647

KEY

Review Problems #1-20

Show the output of the following code segments. If an error occurs, print "error" & explain.

#1
```
String bigNum = "31415";
int littleNum = 45;
System.out.println(bigNum + littleNum);
```
3141545

#2
```
String otherNum = "23";
String otherOtherNum = "32";
String combo = otherNum + otherOtherNum;
String tiny = combo.substring(1, 2);
System.out.println(tiny);
```
combo = "2332" (0 1 2 3)

3

#3
```
int a = 9, b = 6.34;
double sum = a + b;
System.out.println(sum);
```
you can declare multiple variables in one statement

ERROR - 6.34 is not an integer

#4
```
int divA = 19, divB = 4;
double quot = divA / divB;
System.out.println(quot);
```
int division results in 19/4 → 4.75 → 4.0

#5
```
String bestName = "phillies";
int position = bestName.indexOf("m");
int quot = position / 4;
System.out.println(quot);
```
0 1 2 3 ...

→ -1 = position

-1/4 → 0

0

#6
```
int numA = 345, numb = 346;
System.out.println(numb - numA);
```
1

#7
```
String fullName = "Sally Brown";
String firstName = fullName.substring(6);
System.out.println(firstName);
```
0 1 2 3 4 5 6

Brown

#8
```
String number = "5.5";
double myDouble = Double.parseDouble(number);
double total = myDouble + "6.6";
System.out.println(total);
```
converts "5.5" to 5.5 as a double

5.5 + "6.6"

5.56.6

#9
```
int num1 = 6, num2 = 12;
System.out.println(num1 / num2);
```
6/12 → 0.5 → 0

0

#10
```
String answer = "true";
int pos = answer.indexOf("u");
System.out.println(answer + pos);
```
pos = 2

true2

#11
```
String sport1 = "running is the best";
String sport2 = sport1.substring(12);
System.out.println(sport2.substring(2, 2));
```
sport2 = "he best"
Nothing prints

#12
```
double num = Integer.MAX_VALUE + Integer.MIN_VALUE;
System.out.println(num + 2.45);
```
2147483647 + (-2147483648) = -1
-1 + 2.45
-1.45

#13
```
String sport = "running is the best";
String alphabet = "AESZHLXC";
String alphabet2 = alphabet.substring(6);
System.out.println(sport.substring(0, 3) + alphabet2);
```
→ "XC"
"run" "XC"
runXC

#14
```
String num1 = "367921";
int pos1 = num1.indexOf("4");
String num2 = Integer.parseInt(num1);
System.out.println(num2 + pos1);
```
pos1 = -1
num1 = 367921
-1 + 367921
367920

#15
```
int f = 50, h = 10;
double g = 2.5;
double num = f / h - g;
System.out.println(num);
```
50/10 - 2.5 → 5 - 2.5
2.5

#16
```
String name1 = "abcdefgh";
String name2 = name1.substring(3, 3);
System.out.println(name2);
```
needs to be 4 to effectively store a letter
Nothing prints

#17
```
int num1 = 10;
int num2 = num1;
num1 = 15;
System.out.println(num2);
```
num2 = 10
num1 = 15
10

#18
```
int length1 = "Hello World".length();
int length2 = "My name is".length();
System.out.println(length1 + length2);
```
11 = length1
→ 10
11 + 10
21

#19
```
String num1 = "14";
System.out.println(Integer.parseInt(num1) + "28");
```
14 + "28"
1428

#20
```
int num1 = Integer.MAX_VALUE + Integer.MIN_VALUE;
int num2 = num1 + 10;
System.out.println(num2 + 1.4);
```
num1 = -1
-1 + 10 = 9
9 + 1.4
10.4

Nested "For" Loops Practice

#21

```
for(int k = 5; k >= 1; k--)
{
      for (int j = 1; j <= 6; j++)
            System.out.print(j * k + " " );
      System.out.println();
}
```

5 10 15 20 25 30
4 8 12 16 20 24
3 6 9 12 15 18
2 4 6 8 10 12
1 2 3 4 5 6

k	j	j * k + " "
5	1	5
	2	5 10
	3	5 10 15
	4	5 10 15 20
	5	5 10 15 20 25
	6	{5 10 15 20 25 30}
	7 stop!	
4	1	4
	2	4 8
	3	⋮
	4	⋮
	5	⋮
↓	6	{4 8 12 16 20 24}

goes till k = 1

#22

```
for(int k = 9; k >= 2; k--)
{
      for (int j = 2; j <= k; j++)
            System.out.print(k + " " );

      if (k % 2 == 0)
            System.out.println();
}
```

← stops at 2

← if k is even, a new line is created

99999999988888888
777777666666
5555444
332

k	j <= k	prints k + " "
9	2	9
	3	99
	⋮	
	9	999999999
	10 stops!	
8	2	
	⋮	
	8	88888888
	9 stop!	
7	2	
	⋮	
	7	7777777
	8 stop!	

stops when k = 1

#23

```
for(int k = 2; k <=6; k++)
{
      for (int j = 5; j <= 10; j++)
      {
            if ( (j + k) % 2 == 0)
                  System.out.print("AP! ");
            else
                  System.out.print("CS! ");
      }
      System.out.println();
}
```

CS! AP! CS! AP! CS! AP!
AP! CS! AP! CS! AP! CS!
CS! AP! CS! AP! CS! AP!
AP! CS! AP! CS! AP! CS!
CS! AP! CS! AP! CS! AP!

k	j	k + j % 2 == 0?	
2	5	False	CS!
	6	True	CS! AP!
	7	F	
	8	T	
	9	F	
	10	T	{CS! AP! CS! AP! CS! AP!}
	11 stop		
3	5	T	AP!
	6	F	AP! CS!
	↓		
			{AP! CS! AP! CS! AP! CS!}

this repeats for k <= 6

NOTES: Tracing ArrayList Methods

#24

```
ArrayList <String> snacks = new ArrayList<String>();
snacks.add("raisins");        //  raisins
snacks.add("bananas");        //  raisins, bananas
snacks.add("crackers");       //  raisins, bananas, crackers
snacks.add(2, "grapes");      //  raisins, bananas, grapes, crackers
snacks.add("plums");          //  raisins, bananas, grapes, crackers, plums
snacks.remove(0);             //  bananas, grapes, crackers, plums
snacks.set(1, "figs");        //  bananas, figs, crackers, plums
snacks.remove(2);             //
System.out.println(snacks);  // output here:  bananas, figs, plums
```

#25

```
ArrayList <String> snacks = new ArrayList<String>();
snacks.add("raisins");                //
snacks.add("bananas");                //
snacks.add("oreos");                  // raisins, bananas, oreos
String mySnack = snacks.get(1);       // mySnack is: bananas
snacks.add(mySnack);                  // raisins, bananas, oreos, bananas
snacks.set(4, "carrots");             //
System.out.println(snacks.get(0));  // output is:
snacks.remove(0);                                    //
System.out.println(snacks.remove(0));                // output is:
System.out.println(snacks.size());                   // output is:
System.out.println(snacks.get(snacks.size() - 1));// output is:
```

check →

#26

```
ArrayList <String> shoppingCart = new ArrayList<String>();
String box = "Wheaties";
shoppingCart.add(box);            // Wheaties
shoppingCart.add("bananas");      // Wheaties, bananas
shoppingCart.add("crackers");     // wheaties, bananas, crackers
String jar = "Ragu";
shoppingCart.set(1, jar);         // Wheaties, Ragu, crackers
String snack = "Fritos";
shoppingCart.add(2, snack);       // wheaties, Ragu, Fritos, crackers
System.out.println(shoppingCart.remove(0));// output is:  wheaties
String dairy = "milk";            Ragu, Fritos, crackers
shoppingCart.set(1, dairy);
System.out.println(shoppingCart); // output is:  Ragu, milk, crackers
```

#27 **Trace the contents of the array after each line execution.**

```
ArrayList <String> sports = new ArrayList<String>();
sports.add("volleyball");      //  volleyball
sports.add("baseball");        //  volleyball, baseball
sports.add(1, "basketball");   //  volleyball, basketball, baseball
sports.add("soccer");          //  volleyball, basketball, baseball, soccer
sports.remove(2);              //  volleyball, basketball, soccer
sports.set(1, "water polo");   //  volleyball, water polo, soccer
System.out.println(sports);    // output:
```

#28 **Trace the contents of the array after each line execution.**

```
ArrayList <String> desserts = new ArrayList<String>();
desserts.add("cookies");           //
desserts.add("ice cream");         //
desserts.add("brownies");          //  cookies, ice cream, brownies
```

```
String choice1 = desserts.get(1);   //  choice1 → ice cream
desserts.add(choice1);              //  cookies, ice cream, brownies, ice cream
desserts.set(1, "carrot cake");     //  cookies, carrot cake, brownies, ice cream
System.out.print(desserts.get(0));  // output: cookies
desserts.remove(0); results in →    //  carrot cake, brownies, ice cream
System.out.print(desserts.remove(0)); // output: carrot cake
System.out.print(desserts.size());  // output: 2
System.out.print(desserts.get(desserts.size() - 2));  // output: brownies
                                    desserts.get(0)
```

29. **Identify the output of the following code segment.**

```
ArrayList<Integer> list = new ArrayList<Integer>();
list.add(3); list.add(6); list.add(5); list.add(8); list.add(12);
int count=0;
for(int i=0; i<list.size(); i++)
{ if(list.get(i)%2==0)
      count++; }
System.out.println(count);
```

list → 3, 6, 5, 8, 12

if list.get(i) is even then count ++

3

(there are 3 even #'s in the list)

30. **NOTES: Storing Objects in Array Lists & Looping Through Array Lists**

```
public class Grade{
   double gradeAverage; //

   public Grade(double g){ //code not shown  }

   public double getNumericGrade() { //gets numeric grade  }

   public String getLetterGrade(){ //gets letter grade associated with numeric grade }

   public String toString(){  return getNumericGrade() + " = " + getLetterGrade(); }
}
```

Assume the Following Code is written in a Tester Class!!
//instantiate an ArrayList of Grade references (objects)

```
ArrayList<Grade> grades = new ArrayList<Grade>();
```

//write the code to load in 8 random (double) Grade references between 0-100 – use a for loop

```
for (int i=0; i<8; i++)
{
   double randGrade = Math.random()*100;
   grades.add(randGrade);
}
```

//write the code to print out each of the 8 Grades as a letter grade

```
for (int i=0; i<grades.size(); i++)
   S.O.P.(grades.get(i).getLetterGrade());
```

//write the code to print out each of the Grade objects in the ArrayList in reverse order

```
for (int i=grades.size()-1; i>=0; i--)
   S.O.P.(grades.get(i));
```

w/ toString() method each grade will print as 89 = B+, 76 = C-, etc.