Name : _____ Date : _____

ArrayList Worksheet 1 KEY

DIRECTIONS : Fill in each blank with the correct answer/output. Assume each statement happens in order and that one statement may affect the next statement.

```
public class Student{
   private String name;
   private int age;

   public Student(String n, int a){
      name = n;
      age = a;
   }

   public String toString(){
      return name + " is " + age + " years old";
   }
}

ArrayList<Student> rayList = new ArrayList<Student>();
rayList.add(new Student("Sam", 17));
rayList.add(new Student("Sandra", 18));
rayList.add(new Student("Billy", 16));
rayList.add(new Student("Greg", 17));
rayList.add(new Student("Jill", 18));
```

```
System.out.println(rayList.get(0)); // LINE 1

System.out.println(rayList.get(1)); // LINE 2

System.out.println(rayList.get(2)); // LINE 3

System.out.println(rayList.size()); // LINE 4




System.out.println(rayList.remove(0)); // LINE 5

System.out.println(rayList); // LINE 6

System.out.println(rayList.remove(1)); // LINE 7

System.out.println(rayList); // LINE 8
```

```
1.  Sam is 17 years old

2.  Sandra is 18 years old

3.  Billy is 16 years old

4.  5

5.  Sam is 17 years old

6.  [Sandra is 18 years old,
     Billy is 16 years old,
     Greg is 17 years old,
     Jill is 18 years old]

7.  Billy is 16 years old

8.  [Sandra is 18 years old,
     Greg is 17 years old,
     Jill is 18 years old]
```

Name : _____ Date : _____

ArrayList Worksheet 2 KEY

DIRECTIONS : Fill in each blank with the correct answer/output. Assume each statement happens in order and that one statement may affect the next statement.

```
public class Grade{
   //data not shown

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

   public String getLetter(){ //gets letter grade associated with the numeric grade
      //code not shown
   }

   public String toString(){
      return ""+String.format("%.2f",grade);
   }
}


//test code in a client class
//instantiate an ArrayList of Grade references (objects)
ArrayList<Grade> grades = new ArrayList<Grade>();



//write the code to load in 8 random Grade references – use a for loop
for(int i=0; i<8; i++)
{
   grades.add(new Grade(Math.random()*100));
}




//write the code to print out each of the Grades in the ArrayList
for(int i=0; i<grades.size(); i++)
{
   System.out.println(grades.get(i));
}



//write the code to print out each of the 8 Grades as a letter grade
for(int i=0; i<grades.size(); i++)
{
   System.out.println(grades.get(i).getLetter());
}
```

Name : _____ Date : _____

ArrayList Worksheet 3 KEY

Show the output of each block of code below.

1. What is the output?

3

```
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);
```

PART 2 : Fill in the method below with the appropriate code.

```
//this method will return the number
//of times num is present in rayList
public int numCount(ArrayList<Integer> rayList, int num)
{
   int count=0;
   for(int i=0; i<rayList.size(); i++)
   {
      if(rayList.get(i) == num)
        count++;
   }
   return count;
}


//NOTE
//rayList.get(i)==num  is okay due to autobox/autounbox
//It is a bad habit as Java does not autobox/autounbox user defined classes
```

Name : _____ Date : _____

ArrayList Worksheet 4 KEY

Directions : Fill in the method below with the appropriate code.

1.

```
//this method will return the number
//of Strings in rayList with an odd length
public static int countOddLength(ArrayList<String> rayList)
{
   int count=0;
   for(int i=0; i<rayList.size(); i++)
   {
      if(rayList.get(i).length()%2==1)
        count++;
   }
   return count;
}
```

2.

```
//this method will remove all Strings in rayList
//that start with same first letter as firstLetter
public static void removeStrings(ArrayList<String> rayList,
                                 String firstLetter)
{
   for(int i=0; i<rayList.size(); i++)
   {
      if((rayList.get(i).substring(0,1).equals(firstLetter) )
      {
         rayList.remove(i);
         i--;          //necessary so the next element is not skipped
      }
   }
}

// OR

public static void removeStrings(ArrayList<String> rayList,
                                 String firstLetter)
{
   int i=0;
   while(i<rayList.size())
   {
      if((rayList.get(i).substring(0,1).equals(firstLetter) )
         rayList.remove(i);
      else
         i++;
   }
}
```

Name : _____ Date : _____

ArrayList Worksheet 5 KEY

DIRECTIONS : Fill in each blank with the correct answer/output. Assume each statement happens in order and that one statement may affect the next statement.

```
String s = "abcdefghijklmnop";
ArrayList<String> r = new ArrayList<String>();
r.add("abc");
r.add("cde");
r.set(1,"789");
r.add("xyz");
r.add("123");
Collections.sort(r);
r.remove(2);

The first index position in an array is __________.

System.out.print( s.substring(0,1) );     // LINE 2

System.out.print( s.substring(2,3) );     // LINE 3

System.out.print( s.substring(5,6) );     // LINE 4

System.out.print( r.get(0) );             // LINE 5

System.out.print(r.get(0).substring(0,1)); // LINE 6

System.out.print( r.get(2) );             // LINE 7

System.out.print( r.indexOf("123"));      // LINE 8

System.out.print( r.contains("abc"));     // LINE 9

System.out.print( r.isEmpty());           // LINE 10
```

1. 0
2. a
3. c
4. f
5. 123
6. 1
7. xyz
8. 0
9. false
10. false

```
r.set(1, "\\\\");
System.out.print(r);          // LINE 11

r.remove(1);
System.out.print(r);          // LINE 12

r.add("one");
System.out.print(r);          // LINE 13

r.add(0,"five");
System.out.print(r);          // LINE 14

r.clear();
System.out.print(r);          // LINE 15
```

11. [123,\\,xyz]
12. [123,xyz]
13. [123,xyz,one]
14. [five, 123, xyz, one]
15. [ ]