Arrays

1. What is an array in Java?

   An array is a collection of items of the same type.

2. Declare a 2-dimensional array called table of base type int with 4 rows and 5 columns in each row.

   ```java
   int[][] table = new int[4][5];
   ```

3. Write code that will print out the contents of the array you created in Problem 2. Each element should be separated with a space, each row should be on its own line of text.

   ```java
   for (int row = 0; row < table.length; row++)
   {
       for (int col = 0; col < table[row].length; col++)
       {
           if (col != 0)
               System.out.print( " ");
           System.out.print(table[row][col]);
       }
   System.out.println();
   ```

4. Consider the 1-D integer array \{13, 9, 15, 2\}. Show the state of this array after each iteration of the outer loop for bubble and selection sorting algorithms

<table>
<thead>
<tr>
<th>Bubble Sort</th>
<th>Selection Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. 13, 9, 15, 2</td>
<td>0. 13, 9, 15 2</td>
</tr>
<tr>
<td>1. 9, 13, 2, 15</td>
<td>1. 2, 9, 15, 13</td>
</tr>
<tr>
<td>2. 9, 2, 13, 15</td>
<td>2. 2, 9, 15, 13</td>
</tr>
<tr>
<td>3. 2, 9, 13, 15</td>
<td>3. 2, 9, 13, 15</td>
</tr>
<tr>
<td></td>
<td>4. 2, 9, 13, 15</td>
</tr>
</tbody>
</table>
Classes

5. Design a class that represents a personal bank account. Give the class at least two instance variables and three methods. Draw the UML diagram for this class.

<table>
<thead>
<tr>
<th>BankAccount</th>
</tr>
</thead>
<tbody>
<tr>
<td>- ownerName: String</td>
</tr>
<tr>
<td>- amount: double</td>
</tr>
<tr>
<td>- accountId: long</td>
</tr>
</tbody>
</table>

| + getName(): String |
| + setName(String name): String |
| + getAccountId(): long |
| + setAccountId(long id): void |
| + deposit(double depositAmount): void |
| + withdraw(double withdrawAmount): void |
| + earnInterest(double rate): void |
| + getAmount(): double |

6. Write the Java code that defines the class you designed in Problem 5. Include the name of the class, declare the instance variables, and write the methods (at least include the method headers, if you don’t have enough room for the method bodies, leave them empty { /* … */ }).

```java
public class BankAccount
{
    private String ownerName;
    private double amount;
    private long accountId;

    public String getName()
    {
        return ownerName;
    }

    public void setName(String name)
    {
        ownerName = name;
    }

    public long getAccountId()
    {
        return accountId;
    }

    public void setAccountId(long id)
    {
        accountId = id;
    }
}
```
```java
public void deposit(double depositAmount)
{
    amount += depositAmount;
}

public void withdraw(double withdrawlAmount)
{
    if (amount >= withdrawlAmount)
    {
        amount -= withdrawlAmount;
    }
    else
    {
        System.out.println("Insufficient Funds!");
        System.exit(1);
    }
}

public void earnInterest(double rate)
{
    amount *= 1.0 + rate;
}

public double getAmount()
{
    return amount;
}
```