7.1 ArrayList

ArrayList

The Java API provides special classes to store and manipulate groups of objects.
One such class is the **ArrayList**. Standard Java arrays are of a fixed length, which means that after they are created, they cannot expand or shrink.

On the other hand, **ArrayLists** are created with an initial size, but when this size is exceeded, the collection is automatically enlarged.

When objects are removed, the ArrayList may shrink in size. Note that the ArrayList class is in the **java.util** package, so it's necessary to import it before using it.

Create an ArrayList as you would any object:

```
.import java.util.ArrayList;

//...

ArrayList colors = new ArrayList();
```

You can optionally specify a **capacity** and **type** of objects the ArrayList will hold:

```
ArrayList<String> colors = new ArrayList<String>(10);
```

The code above defines an ArrayList of Strings with 10 as its initial size.

Note:

ArrayLists store objects. Thus, the type specified must be a class type. You cannot pass, for example, **int** as the objects' type. Instead, use the special **class types** that correspond to the desired value type, such as **Integer** for int, **Double** for double, and so on.

Q: Drag and drop from the options below to declare an ArrayList to hold 9 Integers.

```
ArrayList<______> ar =
   _______ _________<Integer>(9);
```

ArrayList

The **ArrayList** class provides a number of useful methods for manipulating its objects. The **add**() method adds new objects to the ArrayList. Conversely, the **remove**() methods remove objects from the ArrayList.

Example:

```
import java.util.ArrayList;

public class MyClass {
 public static void main(String[ ] args) {
   ArrayList<String> colors = new ArrayList<String>();
   colors.add("Red");
   colors.add("Blue");
   colors.add("Green");
   colors.add("Orange");
   colors.remove("Green");

   System.out.println(colors);
 }
}
// Output: [Red, Blue, Orange]
```

Note:

Other useful methods include the following:
- **contains**(): Returns true if the list contains the specified element
- **get**(int index): Returns the element at the specified position in the list
- **size**(): Returns the number of elements in the list
- **clear**(): Removes all of the elements from the list

Note: As with arrays, the indexing starts with **0**.

Q: What is the output of this code?

```
ArrayList<String> list = new ArrayList<String>();
```

```
list.add("A");
list.add("B");
list.add("C");
System.out.println(list.get(1));
```

- B
- C
- A