

Multiple Choice

ArrayList Class

Objective 1 –ArrayList Methods

01. Which of the following statement describes *dynamic resizing* as is applies to the **ArrayList** class?

- (A) The size of a new ArrayList object must be specified at instantiation.
- (D) The size of an ArrayList object can be restated throughout a program.
- (C) The size of an ArrayList object is fixed at instantiation.
- (D) The object changes size dynamically as new elements are added.

02. What is the output of this program?

```
public class Java1102
{
    public static void main(String args[])
    {
        ArrayList names = new ArrayList();
        names.add("Isolde");
        names.add("John");
        names.add("Greg");
        names.add("Maria");
        names.add("Heidi");
        System.out.println("names contains " + names);
    }
}
```

- (A) names contains [Isolde, John, Greg, Maria, Heidi]
- (B) names contains [Heidi, Maria, Greg, John, Isolde]
- (C) names contains [John, Greg, Maria, Heidi]
- (D) names contains [Isolde, John, Greg, Maria]

03. What is the output of this program segment?

```
ArrayList names = new ArrayList();
names.add("John");
names.add("Greg");
names.add("Maria");
names.add("Heidi");
names.add(2, "Jessica");
System.out.println();
for (int k = 0; k < names.size(); k++)
    System.out.print(names.get(k) + " ");
```

- (A) John Greg Jessica Heidi
- (B) John Jessica Maria Heidi
- (C) John Jessica Greg Maria Heidi
- (D) John Greg Jessica Maria Heidi

04. What is the output of this program segment?

```
ArrayList names = new ArrayList();
names.add("John");
names.add("Greg");
names.add("Maria");
names.add("Heidi");
names.set(2, "Jessica");
System.out.println();
for (int k = 0; k < names.size(); k++)
    System.out.print(names.get(k) + " ");
```

- (A) John Greg Jessica Heidi
- (B) John Jessica Maria Heidi
- (C) John Jessica Greg Maria Heidi
- (D) John Greg Jessica Maria Heidi

05. Consider the following code segment.

```
ArrayList names = new ArrayList();
names.add("John");
names.add("Greg");
names.add("Maria");
names.add("Heidi");
names.remove(1);
names.remove(2);
System.out.println();
for (int k = 0; k < names.size(); k++)
    System.out.print(names.get(k) + " ");
```

What is printed as a result of executing the code segment?

- (A) John Maria
- (B) John Heidi
- (C) Greg Heidi
- (D) Greg Maria

06. Consider the following code segment.

```
ArrayList names = new ArrayList();
names.remove(1);
names.remove(2);
names.add("John");
names.add("Greg");
names.add("Maria");
names.add("Heidi");
System.out.println();
for (int k = 0; k < names.size(); k++)
    System.out.print(names.get(k) + " ");
```

What is printed as a result of executing the code segment?

- (A) John Maria
- (B) John Heidi
- (C) Greg Heidi
- (D) Greg Maria
- (E) Error

07. What is the output of this program segment?

```
ArrayList names = new ArrayList();
names.add("Isolde");
names.add("John");
names.add("Greg");
names.add("Maria");
names.add("Heidi");
for (int k =0; k < 5; k++)
    names.add(k,"Jessica");
System.out.println(names);
```

- (A) [Jessica, Isolde, Jessica, John, Jessica, Greg, Jessica, Maria, Jessica, Heidi]
- (B) [Isolde, John, Greg, Maria, Heidi, Jessica, Jessica, Jessica, Jessica, Jessica]
- (C) [Jessica, Jessica, Jessica, Jessica, Jessica, Isolde, John, Greg, Maria, Heidi]
- (D) [Isolde, Jessica, John, Jessica, Greg, Jessica, Maria, Jessica, Heidi, Jessica]
- (E) Error

08. What is the output of this program segment?

```
ArrayList names = new ArrayList();
names.add("Isolde");
names.add("John");
names.add("Greg");
names.add("Maria");
names.add("Heidi");
for (int k =0; k < 5; k++)
    names.add(k,"Jessica");
System.out.println(names);
```

- (A) [Jessica, Isolde, Jessica, John, Jessica, Greg, Jessica, Maria, Jessica, Heidi]
- (B) [Jessica, Jessica, Jessica, Jessica, Jessica]
- (C) [Isolde, John, Greg, Maria, Heidi]
- (D) [Isolde, Jessica, John, Jessica, Greg, Jessica, Maria, Jessica, Heidi, Jessica]
- (E) Error

Objective 2 – ArrayList and Java Primitive Types

09. What is the output of this program?

```
public class Java1109
{
    public static void main(String args[ ])
    {
        ArrayList numbers = new ArrayList();
        numbers.add(new Integer(11));
        numbers.add(new Integer(22));
        numbers.add(new Integer(33));
        numbers.add(new Integer(44));
        numbers.add(new Integer(55));
        System.out.println(numbers);
    }
}
```

- (A) 11
22
33
44
55
- (B) 55
- (C) 11 22 33 44 55
- (D) [11, 22, 33, 44, 55]
- (E) Error

10. Java provides *wrapper* classes, which create objects that store primitive data types. Which of the following are Java *wrapper* classes?

- (A) int, double, boolean, string
- (B) int, double, boolean
- (C) Integer, Double, Boolean
- (D) integer, real, logic
- (E) Integer, Double, Boolean, String

11. Which of the following statements correctly uses a *wrapper* class to store a primitive data type?

- (A) `intList.add(new Integer(1000));`
- (B) `doubleList.add(new Double(123.321));`
- (C) `logicList.add(new Boolean(true));`
- (D) All of the above.

12. What is the output of this program?

```
import java.util.ArrayList;

public class Java1112
{
    public static void main(String args[ ])
    {
        ArrayList numbers = new ArrayList();

        int k;
        for (k = 1; k <= 10; k++)
            numbers.add(new Integer(k));

        int sum = 0;
        for (k = 0; k < numbers.size(); k++)
        {
            Integer temp = (Integer) numbers.get(k);
            sum += temp.intValue();
        }

        double average = (double) sum / numbers.size();
        System.out.println(average);
    }
}
```

- (A) 5
- (B) 5.5
- (C) 10
- (D) 50
- (E) 55

Objective 4 – ArrayList and Generics

13. What is known by the declaration `ArrayList list = new ArrayList();` ?

- (A) `list` is an `ArrayList` object.
- (B) Elements of the `list` array are objects.
- (C) The type of information stored by `list` is unknown.
- (D) The number of array elements is not specified.
- (E) All of the above

14. What is known by the declaration `ArrayList<String> list = new ArrayList<String>();` ?

- (A) `list` is an `ArrayList` object.
- (B) Elements of the `list` array are objects.
- (C) The type of objects stored by `list` are `Integer` objects.
- (D) The number of array elements is not specified.
- (E) All of the above

15. What is guaranteed by a generic declaration like the one shown below?

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

- (A) `ArrayIndexOutOfBoundsException` error will not happen.
- (B) There will not be any compile errors due to wrong data types.
- (C) At execution time every element of `list` will store a `String` object.
- (D) Improper access to any `list` member is not possible.
- (E) All of the above

16. Consider the following program segment.

```
ArrayList<Double> reals = new ArrayList<Double>();  
list2.add(400.0);  
list2.add(500.0);  
list2.add(600.0);
```

Which of the following statements demonstrates the use of generics?

- (A) **Double real = reals.get(0);**
- (B) **double real = (reals.get(0)).doubleValue();**
- (C) **double real = ((Double)reals.get(0)).doubleValue();**
- (D) **Double real = ((Double)reals.get(0));**
- (E) Both A & B

17. Consider the following **Person** class.

```
class Person  
{  
    private String name;  
    private int age;  
  
    public Person (String n, int a)  
    {  
        name = n;  
        age = a;  
    }  
}
```

Which of the following statements correctly declares a generic **ArrayList** object of Person objects?

- (A) **ArrayList<Person> people = new ArrayList<Person>();**
- (B) **ArrayList<Person> people = new Person();**
- (C) **Person people = new ArrayList();**
- (D) **ArrayList people = new ArrayList(Person);**
- (E) **ArrayList<Person> people = new ArrayList();**

18. What is the output of the following program segment?

```
ArrayList<String> people = new ArrayList<String>();  
people.add("Kathy Coleman");  
people.add("Tom Terrific");  
System.out.println(people);
```

- (A) Kathy Coleman
Tom Terrific
- (B) Tom Terrific
Kathy Coleman
- (C) [Kathy Coleman, Tom Terrific]
- (D) Error

19. What is the output of the following program segment?

```
ArrayList people = new ArrayList();  
people.add("Kathy Coleman");  
people.add(new Integer(1000));  
System.out.println(people);
```

- (A) Kathy Coleman
1000
- (B) 1000
Kathy Coleman
- (C) [Kathy Coleman, 1000]
- (D) Error

20. What is the output of the following program segment?

```
ArrayList<String> people = new ArrayList<String>();  
people.add("Kathy Coleman");  
people.add(new Integer(1000));  
System.out.println(people);
```

- (A) Kathy Coleman
1000
- (B) 1000
Kathy Coleman
- (C) [Kathy Coleman, 1000]
- (D) Error

Objective 5 – ArrayList and the Enhanced For Loop

21. Consider the following program segment.

```
ArrayList<String> names = new ArrayList<String>();  
names.add("Isolde");  
names.add("John");  
names.add("Greg");  
names.add("Maria");
```

Which of the following statements display all the elements in the **names** array?

- (A) **System.out.println(names);**
- (B) **for (int index = 0; index < names.size(); index++)
System.out.println(names.get(index));**
- (C) **for (String name: names)
System.out.println(name);**
- (D) All of the above

22. Assume the following declaration.

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

Which of the following statements stores "Kathy" in the **list** array:

- (A) `list.add("Kathy");`
- (B) `for (String item: list)
 item = "Kathy";`
- (C) `list[10] = "Kathy";`
- (D) All of the above

23. Consider the following program segment.

```
ArrayList<String> names1 = new ArrayList<String>();  
ArrayList<String> names2 = new ArrayList<String>();
```

```
names1.add("Isolde");  
names1.add("John");  
names1.add("Greg");  
names1.add("Maria");  
names1.add("Heidi");
```

```
for (String name: names1)  
    names2.add(name);
```

Which of the following statements describes the correct execution of the program segment?

- (A) The segment cannot execute due to a compile error.
- (B) The elements of `names1` are copied into `names2` in reverse order.
- (C) The elements of `names1` are copied into `names2` in the same order.
- (D) The elements of `names2` are copied into `names1` in the same order.

24. Consider the following program segment.

```
ArrayList<String> names1 = new ArrayList<String>();
ArrayList<String> names2 = new ArrayList<String>();
names1.add("Isolde");
names1.add("John");
names1.add("Greg");
names1.add("Maria");
names1.add("Heidi");
for (String name: names2)
    names1.add(name);
System.out.println(names1);
```

What is the output as a result of executing the program segment?

- (A) [Isolde, John, Greg, Maria, Heidi]
- (B) [Isolde, John, Greg, Maria, Heidi, Isolde, John, Greg, Maria, Heidi]
- (C) []
- (D) The segment cannot execute due to a compile error.

25. Consider the following program segment.

```
ArrayList<String> names = new ArrayList<String>();
names.add("Isolde");
names.add("John");
names.add("Greg");
names.add("Maria");
names.add("Heidi");
for (String name: names)
    names.add(name);
System.out.println(names) ;
```

What is the output as a result of executing the program segment?

- (A) [Isolde, John, Greg, Maria, Heidi]
- (B) [Isolde, John, Greg, Maria, Heidi, Isolde, John, Greg, Maria, Heidi]
- (C) Compile error message
- (D) Program crashes during execution with a runtime exception errors.

26. Consider the following program segment.

```
int[] list1 = {1,2,3,4,5,6,7,8,9};  
ArrayList<Integer> list2 = new ArrayList<Integer>();
```

Which of the following code segments copies the elements from **list1** into **list2** ?

I. **for (int k = 0; k < list1.length; k++)**
 list2.add(list1[k]);

II. **for (Integer number: list2)**
 list1.add(number);

III. **for (int number: list1)**
 list2.add(number);

- (A) I only
- (B) II only
- (C) III only
- (D) I and II only
- (E) I and III only

Objective 6 – Two Dimensional Dynamic Arrays

27. Which of the following declares **mammals** as a two-dimensional dynamic array?

- (A) `ArrayList<ArrayList<String>> mammals = new ArrayList<ArrayList<String>>();`
- (B) `ArrayList<String,String> mammals = new ArrayList<String,String>();`
- (C) `ArrayList<String><String> mammals = new ArrayList<String><String>();`
- (D) `ArrayList<ArrayList<String,String>> mammals = new ArrayList<ArrayList<String,String>>();`
- (E) All of the above

28. Assume that **mammals** is correctly declared as a two-dimensional dynamic array of **String** elements.

Which of the following will display every element of the **mammals** array?

I.

```
for (ArrayList<String> mammal: mammals)
{
    for (String animal: mammal)
        System.out.println(animal);
}
```

II.

```
System.out.println(mammals);
```

III.

```
for (row = 0; row < mammals.size(); row++)
{
    for (int col = 0; col < mammals.get(row).size(); col++)
        System.out.println(mammals.get(row).get(col));
}
```

(A) I only

(B) I and II only

(C) I and III only

(D) II and III only

(E) I, II and III