

Part I: Multiple Choice (12 questions / 4 points each)

- 1) What is the difference between a regular instance field and a static instance field?
- 2) What is the difference between a regular instance field and a static instance field?
- 3) What advantages does using an ArrayList have over using an array?
- 4) Know what the add & set methods of the ArrayList class do.
- 5) What advantages does using an array have over using an ArrayList?
- 6) Know what the get & remove methods of the ArrayList class do.
- 7) Know how to transfer the contents of an array of Strings to an ArrayList of Strings.
- 8) Suppose an ArrayList of Strings contains the individual words of a sentence. Be able to write a method that would remove, from the ArrayList, all occurrences of the word passed to the method.
- 9) Be able to debug code that is using an ArrayList of user created objects.
- 10) Given a class with a parametric constructor, be able to construct an instance of that class.
- 11) Given two classes each containing a parametric constructor. If the second class has an instance field of the type of the first class, be able to construct an instance of the second class.
- 12) Given an ArrayList of objects of the second class from exercise 11 above. Be able to call the appropriate methods to find the data requested.

Samples:

- 1) Consider the following code segment, applied to list, an ArrayList of Integer values.

```
int len = list.size();

for (int j=0; j<len; j++)
{
    list.add(j+1, new Integer(2*j));
    list.set(j, new Integer(j+3));
}
```

If list is initially 4 9 3, what will it be following execution of the code segment?

- a) 3 0 2 4 9 3
 - b) 3 4 5 4 9 3
 - c) 3 4 5 0 4 9 6 3
 - d) 3 4 4 9 9 9
 - e) 3 0 9 3
- 2) Which of the following is **NOT** an advantage of ArrayLists over Arrays? (Circle two)
 - a) easier to insert data into an ArrayList
 - b) easier to delete data from an ArrayList
 - c) easier to retrieve data from an ArrayList
 - d) easier to resize an ArrayList
 - e) easier to overwrite an element of an ArrayList

For questions 3 - 5, use the Grade and Student classes listed below.

```
public class Grade
{
    private int hwAve;
    private int testAve;
    private String letter;

    public Grade(int hw, int test)
    {
        hwAve = hw;
        testAve = test;
        setLetter();
    }
    private void setLetter()
    {
        double ave = 0.2 * hwAve + 0.8 * testAve;

        if (ave > 89) letter = "A";
        if (ave > 79 && ave < 90) letter = "B";
        if (ave > 69 && ave < 80) letter = "C";
        if (ave > 59 && ave < 70) letter = "D";
        if (ave < 60) letter = "F";
    }
    public int gethwAve()
    {
        return hwAve;
    }
    public int gettestAve()
    {
        return testAve;
    }
    public String getLetter()
    {
        return letter;
    }
}
```

```
public class Student
{
    private String name;
    private Grade grd;

    public Student(String n, Grade g)
    {
        name = n;
        grd = g;
    }
    public String getName()
    {
        return name;
    }
    public Grade getGrade()
    {
        return grd;
    }
}
```

3) Which of the following correctly creates a Grade Object called x?

- a) `Grade x = new Grade(96, 85, "B");`
- b) `Grade x = new Grade(96, 85, B);`
- c) `Grade x = new Grade("96", "85", "B");`
- d) `Grade x = new Grade("96", "85");`
- e) `Grade x = new Grade(96, 85);`

4) Which of the following correctly creates a Student Object called y?

- a) `Student y = new Student("Bob", "A");`
- b) `Student y = new Student(Bob, A);`
- c) `Student y = new Student(Bob, new Grade(98, 91));`
- d) `Student y = new Student("Bob", new Grade(98, 91));`
- e) `Student y = new Student("Bob", new Grade(98, 91, "A"));`

5) Consider the partial AllStudents class listed below:

```
public class AllStudents
{
    private ArrayList<Student> stuList = new ArrayList<Student>();

    //Default constructor fills stuList from a text file.
    public AllStudents()
    {
        ...
    }
    //Public member method is passed the name of a Student and returns
    //the testAve of that Student.
    public int locate(String nme)
    {
        for (int j=0; j<stuList.size(); j++)
        {
            <code segment>
        }
        return null; // name not found
    }
}
```

Which of the following could replace <code segment> so that the method would work as intended?

- I.

```
if (stuList[j].getName().equals(nme))
{
    Grade x = stuList[j].getGrade();
    return x.gettestAve();
}
```
- II.

```
Student x = stuList.get(j);
if (x.getName().equals(nme))
{
    Grade y = x.getGrade();
    return y.gettestAve();
}
```
- III.

```
Student x = stuList.get(j);
if (x[j].getName().equals(nme))
{
    Grade y = x[j].getGrade();
    return y.gettestAve();
}
```

- a) I only b) II only c) III only d) II and III only e) I, II, and III

Part II: Free Response - complete the code, labeled <to be completed by you>, in the StockItem, Store, and Driver classes.

```
public class StockItem
{
    private String name;
    private int idnum;
    private double price;
    private int numOnShelf;

    public StockItem(String n, int id, double p, int num)
    {
        name = n;
        idnum = id;
        price = p;
        numOnShelf = num;
    }
    public String getName()
    {
        return name;
    }
    public int getID()
    {
        return idnum;
    }
    public double getPrice()
    {
        return price;
    }
    public int getNum()
    {
        return numOnShelf;
    }
    //This method changes the price of an item
    public void setPrice(double p)
    {
        <to be completed by you>
    }
    //This method removes n items from the number already on the shelf.
    //If an attempt is made to remove more than the number on the shelf,
    //all are removed.
    public void remove(int n)
    {
        <to be completed by you>
    }
    //This method adds n items to the number already on the shelf.
    public void add(int n)
    {
        <to be completed by you>
    }
    public String toString()
    {
        if (name.length() >= 8)
            return (idnum + "\t" + name + "\t$" + price + "\t" + numOnShelf);
        else
            return (idnum + "\t" + name + "\t\t$" + price + "\t" + numOnShelf);
    }
}
```

```

import java.util.ArrayList;

public class Store
{
    private ArrayList<StockItem> myStockList;

    public Store()
    {
        myStockList = new ArrayList<StockItem>();
    }
    //This method adds a StockItem to the ArrayList
    public void add(StockItem s)
    {
        <to be completed by you>
    }
    //This method searches for the StockItem in myStockList whose id number
    //matches idnum and removes all of that item from the shelf. You may
    //assume idnum does exist. The method returns the product of the number
    //of items removed from the shelf and the price of each.
    public double removeAll(int idnum)
    {
        <to be completed by you>
    }
    //This method replaces StockItem x with StockItem y in the ArrayList
    //myStockList.
    public void replace(StockItem x, StockItem y)
    {
        <to be completed by you>
    }
    //This method prints out the contents of the ArrayList.
    public void list()
    {
        System.out.println("ID\tName\t\tPrice\tQuantity");
        System.out.println("--\t----\t\t\t-----\t-----");

        <the rest to be completed by you>
    }
}

public class Driver
{
    public static void main(String[] args)
    {
        Store inventory = new Store();

        StockItem a = new StockItem("Special K", 1001, 2.35, 12);
        inventory.add(a);

        StockItem b = new StockItem("Kleenix", 1002, 1.76, 8);
        inventory.add(b);

        StockItem c = new StockItem("Coke", 1003, 1.49, 15);
        inventory.add(c);

        StockItem d = new StockItem("Peanuts", 1004, 2.29, 6);
        inventory.add(d);

        StockItem e = new StockItem("Soap", 1005, 0.89, 25);
        inventory.add(e);
    }
}

```

```
//Print the contents of inventory
<to be completed by you>

//Increase the price of Soap to 0.92
<to be completed by you>

//Remove 6 boxes of Kleenix from the shelf
<to be completed by you>

//Add 24 6-packs of Coke to the shelf
<to be completed by you>

//The peanuts are stale, remove all of them from the shelf
<to be completed by you>

//Print the value of the peanuts that were removed from the shelf
<to be completed by you>

//Replace "Special K" with ("Total", 1006, 2.24, 48)
<to be completed by you>

//Print the contents of inventory
<to be completed by you>
    }
}
```

Java Concepts Review Assignment:

Pages 324 - 327

Exercises: R7.1, R7.3, R7.4, R7.14, R7.16d,
R7.18c, R7.18d