

First Name: \_\_\_\_\_

AITI 2004: Exam 1 – June 30, 2004

Last Name: \_\_\_\_\_

Circle One: OCW Non-OCW

**Read Instructions Carefully!**

This is a 3 hour closed book exam. No calculators are allowed. Please write clearly – if we cannot understand your answer, you will receive 0 points for that question. This exam has 6 parts worth a total of 100 points.

**Section 1 – Java Basics (20 points)**

1. (2 points) ( T / F ) Java programs must be compiled separately for each platform.

2. (2 point) ( T / F ) In Java, the identifiers Bob and bob refer to the same variable.

3. (2 points) The command we use to compile Java code is

- a. java
- b. javac

4. (2 points) The command we use to run a Java program is

- a. java
- b. javac

5. (2 points) Which of the following are valid variable names?

- a. true6
- b. you-me
- c. hello Java
- d. DEGREE

6. (3 points) Which of the following are valid statements?

- a. `int i = 5`
- b. `short s = 5;`
- c. `boolean firstTime;`
- d. `char c = '4';`

7. (2 points) What is the value of the following arithmetic expressions?

- a.  $(4 + 3) * 6 - 2 = \underline{\hspace{2cm}}$
- b.  $14 \% (18 / 5) = \underline{\hspace{2cm}}$

8. (5 points) What is the value of the following Boolean expressions?

```
boolean b1 = false;
boolean b2 = true;
```

- a. `b1 && b2 = \underline{\hspace{2cm}}`
- b. `b1 || b2 = \underline{\hspace{2cm}}`
- c. `b2 || b1 = \underline{\hspace{2cm}}`
- d. `!b1 = \underline{\hspace{2cm}}`
- e. `(!b1) && b2 = \underline{\hspace{2cm}}`

## Section 2: Control Structures (15 points)

1. (2 points) What does the following code print to the screen?

```
int score = 75;
String orange = "apple";
String apple = "orange";

if (score >= 60)
    System.out.println(apple);
else
    System.out.println(orange);
```

a. apple            b. orange

2. (2 points) What is the output of the following code?

```
char grade = 'D';
switch (grade) {
    case 'A':
        System.out.println("Excellent");
        break;
    case 'B':
        System.out.println("Good");
        break;
    case 'C':
        System.out.println("OK");
        break;
    default:
        System.out.println("Let's talk");
}
```

3. (5 points) Write a for loop that calculates the sum of the first 100 multiples of 3, (3, 6, 9, ...). (Note: This is not the same as the sum of the multiples of 3 from 1 to 100)

4. What is the output for the following segment of the code?

a) (3 points)

```
int k = 10;
while (k < 10) {
    k += 2;
    System.out.println(k);
}
```

b) (3 points)

```
int k = 10;
while (k < 20) {
    k += 2;
    System.out.println(k);
}
```

### **Section 3: Methods and Arrays (10 points)**

1. (5 points) Write a method, `product`, which takes an integer array as an argument and returns the product of the numbers in that array. (It multiplies the elements of the array).

2. (5 points) The following piece of code calls a method called `scaleProduct`. The first argument to `scaleProduct` is an array of integers and the second is an integer. The method takes the product of the array and returns the sum of the product and the second argument.

```
int[] numbers = {1,2,3,1};
int scale = 3;
int ans = scaleProduct(numbers, scale);           //1*2*3*1 + 3
System.out.print(ans);
```

Output:

9

Write the `scaleProduct` method. To take the product of the array, it should call the `product` method you wrote above.

```
public int scaleProduct(){

}

}
```

## Section 4 – Lists and Iterators (15 points)

1.(4 points) List one advantage and one disadvantage of using an `ArrayList` compared to an array.

2. (4 points) Would an `ArrayList` or a `LinkedList` be more appropriate for the following applications?

- To store the people waiting in line at a bank, where a lot of people are continually joining the line at the end and leaving from the front, but you never want to search or reorder the line.

a) `ArrayList`

b) `LinkedList`

- To store the grade books of a fixed number of students in this class that you want to update.

a) `ArrayList`

b) `LinkedList`

3. (7 points) Using an `Iterator`, complete the method below that takes the product of all the integers in the given `ArrayList`. You can assume `java.util.*` is imported. (Hint: The `Integer` class has a method with signature: `public int intValue()`)

```
static int product(ArrayList intList) {
```

```
}
```

### **Section 5 – Packages Access and Scope (8 points)**

```
class TestScope {
    int x = 0;
    int y = 0;

    void f() {
        int y; //LINE A
        y = 20;
        x = 10;
    }

    void print() {
        System.out.println(x);
        f();
        System.out.println(x);
        System.out.println(y);
    }
}
```

1. (4 points) What will this code print out?
2. (4 points) What will this code print out if we remove LINE A?

### **Section 5 – Static and Final (12 points)**

1. (4 points) Add two public constants of type `double` to the `Thermometer` class below, one for the freezing point and one for the boiling point in Celsius. The freezing point is 0 degrees Celsius and the boiling point is 100 degrees Celsius.
2. (4 points) Add a public static method to the `thermometer` class to convert degrees Celsius to degrees Fahrenheit. It should accept a `double` argument in Celsius and return a `double` in Fahrenheit. The conversion is  $Fahrenheit = 1.8 * Celcius + 32$ .

```
package thermo;

public class Thermometer {

}
```

3. (4 points) In the main method of `UseThermometer`, use the `Thermometer` class to calculate and print the boiling point **in degrees Fahrenheit**.

```
package thermo;

public class UseThermometer {

    public static void main(String args[]) {

    }

}
```

### **Section 6 – Classes (20 points)**

1. On the next page write a class to represent a football team in a package called `football`. We want to be able to access the class `FootballTeam` from **any** package.

The team has the following properties, which **cannot** be accessed outside of the class.

- name of the team
- number of wins
- number of losses

Write a constructor that accepts the name of the team, the number of wins, and the number of losses as arguments and sets the class properties to those values. This constructor should be accessible from **any** package.

Write a second constructor that takes only the name of the team as an argument. This constructor should set the name of the team to the argument and set the number of wins and losses to 0. (Hint: The body of this constructor should be one line and it should call the first constructor.) This constructor should be accessible from **any** package.

Write methods that return the name of the team, the number of the wins, and the number of losses. These methods should be accessible from **any** package.

Next write a method to increase the numbers of wins by 1 and another method to increase the number of losses by one. These methods should only be accessible from **inside** the `football` package.

Write a method that returns true when a team has a "good" record, meaning the team has more wins than losses. This method should be accessible from **any** package.

Finally, add a `main` method to the `FootballTeam` class. In the `main` method, construct a `FootballTeam` named "AITI" with 3 wins and 5 losses. Call the method that returns true when the team has a good record and print out the result. Now make three calls to the method that increases the number of wins by 1. Lastly, call the "good record" method again and print out the result.



