University of Maryland College Park  
Dept of Computer Science  
CMSC131 Fall 2015  
Midterm I Key

Last Name (PRINT): ____________________________________________

First Name (PRINT): ____________________________________________

University Directory ID (e.g., umcpturtle)___________________________

I pledge on my honor that I have not given or received any unauthorized assistance on this examination.

Your signature: ________________________________________________

Instructions

- This exam is a closed-book and closed-notes exam.
- Total point value is 200 points.
- The exam is a 50 minutes exam.
- Please use a pencil to complete the exam.
- WRITE NEATLY.
- There are four problems in the exam.
- You don’t need to use meaningful variable names; however, we expect good indentation.

Grader Use Only

<table>
<thead>
<tr>
<th>#1</th>
<th>Problem #1 (Short Answers)</th>
<th>(76)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#2</td>
<td>Problem #2 (Conditionals)</td>
<td>(44)</td>
</tr>
<tr>
<td>#3</td>
<td>Problem #3 (Loops)</td>
<td>(44)</td>
</tr>
<tr>
<td>#4</td>
<td>Problem #4 (Coding)</td>
<td>(36)</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>Total</strong></td>
<td>(200)</td>
</tr>
</tbody>
</table>

NOTES:

1. Xiyang will be the lead grader. You need to contact him if you want to change anything about the key.
2. Teaching TAs: Pick one random exam and photocopy it before returning your exams. This will allow us to post a note that we photocopy selected exams.
3. Please enter the grades in the grades server. Notice we have entries for each question.
4. Grading Deadline: Monday Oct 19. The midterm will be returned in lab that day.
5. Xiyang: On Monday Oct 19 verify the scores have been entered and drop us a line so we can make the grades visible.
Problem #1 (Short Answers)

1. (4 pts) Java compilers produce:
   a. Assembly code.
   b. A third-generation code.
   c. Bytecode.
   d. Pure machine code (can run on the computer CPU).
   e. None of the above.
   **ANSWER: C**

2. (4 pts) How many different combinations of 0’s and 1’s can be represented with 5 bits?
   **ANSWER: 32**

3. (4 pts) Which of the following are reserved names in Java? Circle all that apply.
   a. position
   b. else
   c. location
   d. if
   e. None of the above
   **ANSWER: b and d**

4. (4 pts) Which of the following expressions are equivalent based on the two String variables s and t? Circle all that apply.
   ```java
   String s = "Mary";
   String t = "Mary";
   a. s.equals(t)
   b. s.compareTo(t) == 0
   c. None of the above
   ```
   **ANSWER: a and b**

5. (4 pts) Name two primitive types used to store integers in addition to int and long.
   **ANSWER: byte and short**

6. (4 pts) What value would Java compute for the following Java code fragments?
   a) int x = 1; int y = x / 2; // What is the value of y?
   **ANSWER:**
   b) int w = 15 % 3; // What is the value of w?
   **ANSWER:**

7. (4 pts) Which of the following could be used to name variables in Java? We’re not asking if they are good style, just whether or not they are permissible. Circle all that apply.
   _environment#        surface10         10CaRdS       salt&pepper
   **ANSWER: surface10**
8. (8 pts) Re-write (in the box) the following code fragment using a for-loop. The body of the for loop can only have the System.out.println statement (no other statement).

```java
int y = 3, val = 10;
while (y >= 1) {
    System.out.println("val: "+val);
    val*=2;
    y--;
}
```

**ANSWER:**

```java
for (int y = 3, val = 10; y >= 1; val *= 2, y--)
{
    System.out.println("val: "+val);
}
```

9. (4 pts) Write the binary representation of 17.

**ANSWER: 10001**

10. (4 pts) Write the decimal equivalent of the binary number 11011.

**ANSWER: 27**

11. (4 pts) What is pseudocode?

**ANSWER:** English-like description of the set of steps required to solve a problem

12. (4 pts) What is null?

**ANSWER:** A Java value meaning no address

13. (4 pts) Complete the following assignment so we are able to print the following message. Notice that double quotes surround the message.

```
"C:\home\tmp"
```

**ANSWER:** "\"C:\home\tmp\""

14. (4 pts) How many distinct String object instances are created in the following code segment?

```java
String movie = "The Martian";
String november = new String("Hunger Games");
String best = november;
```

**ANSWER: 2**

15. (4 pts) Define a String constant named BEST_SCHOOL that has as value “UMCP”.

**ANSWER:** final String BEST_SCHOOL = "UMCP"
16. (4 pts) Write the output generated by the following statements.

```java
int y = 10;
int x = y++;  
System.out.println(y);  
System.out.println(x);  
```

**ANSWER:** y is 11 and x is 10.

17. (4 pts) Will the value of x change in the following code? Briefly explain (yes or no answer with no explanation will receive no credit)

```java
int x = 20, y = 10;  
if ( (y >= 10) || (++x > 20)) {  
}  
```

**ANSWER:** No because of short circuit

18. (4 pts) The following code fragment generates an error when run. Why?

```java
String k = null;  
int x = k.length();  
```

**ANSWER:** K is not an object
**Problem #2 (Conditionals)**

Fill in the method below in order to complete a program called **Dessert**. The program reads a number of calories and prints the dessert classification based on the table below.

<table>
<thead>
<tr>
<th>Number of Calories</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than or equal to a 100</td>
<td>Awesome</td>
</tr>
<tr>
<td>More than 100 and less 200</td>
<td>OK</td>
</tr>
<tr>
<td>Greater than or equal to 200</td>
<td>Danger!</td>
</tr>
</tbody>
</table>

**Restrictions/Assumptions**

- Use the message “Enter calories: ” to read the number of calories.
- You must use `System.out.println` to print the classification.
- You should use the Scanner class to read values.

```java
public class Dessert {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter calories: ");
        String classification;
        int calories = scanner.nextInt();
        if (calories <= 100) {
            classification = "Awesome";
        } else if (calories > 100 && calories < 200) {
            classification = "OK";
        } else {
            classification = "Danger!";
        }
        System.out.println(classification);
        scanner.close();
    }
}
```
Problem #3 (Loops)

Fill in the method below in order to complete a program called AccountAccess. The program will keep asking for a login id (string value) and a pin number (integer) as long as the values provided are different than “Bob” and 1234, respectively. Each time invalid values are provided the program will print the message “Invalid values.” Once “Bob” and 1234 are provided the program will display the message “Access Granted” and end. For this problem:

- Use the Scanner class to read values.
- Use System.out.println to display the messages “Enter login id: “ and “Enter pin number: .“
- Use the JOptionPane.showMessageDialog method to display “Invalid values” and “Access Granted.”
- You must use a do while statement.

```java
public class AccountAccess {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        boolean valid;

        do {
            System.out.println("Enter login id: ");
            String loginid = scanner.next();
            System.out.println("Enter pin number: ");
            int pinNumber = scanner.nextInt();
            valid = loginid.equals("Bob") && pinNumber == 1234;
            if (!valid) {
                JOptionPane.showMessageDialog(null, "Invalid values");
            }
        } while (!valid);
        JOptionPane.showMessageDialog(null, "Access Granted");
        scanner.close();
    }
}
```
Problem #4 (Coding)

Fill in the method below. The method will display a diagram where each row has two more * than the previous row. The number of rows corresponds to the size parameter. Notice the first row will have two *. The following are examples of diagrams the method will generate:

For a size parameter of 2:
**
****

For a size parameter of 3:
**
****
******

For a size parameter of 4:
**
****
******
*******

Your solution must handle different size values (not just 2, 3, and 4).

```java
public static void printDiagram(int size) {
    int row, col;

    for (row = 1; row <= size; row++) {
        for (col = 1; col <= row; col++) {
            System.out.print("**");
        }
        System.out.println();
    }
}
```