

Sample CSE8A midterm
Multiple Choice (circle one)

(2 pts) Evaluate the following Boolean expressions and indicate whether “short-circuiting” happened during evaluation: Assume variables with the following names and values:

```
int moos = 49;  
int quacks = 8;  
int temp = -5;
```

Expression 1:

```
((quacks % 3) == 2) && (temp > -5) || (moos == (quacks+41))
```

1. The Boolean expression above evaluates to which?

true false

2. In the evaluation of that Boolean expression, did ANY short-circuiting of evaluation occur?

YES NO

3. (1pt) What does the solitary “.” stand for in the Unix command below?
Choose the **most precise** answer.

```
cp MyStuff/J.java .
```

- (a) The current file.
- (b) The destination file.
- (c) The home directory.
- (d) The current directory.
- (e) A file created for our Lab classes.
- (f) The directory above the current directory.
- (g) The directory below the current directory.

4. (2pts) How many lines of output would be printed by the following code?

```
for (int i = 0; i <= 40; i++)  
    for (int j = 0; j < 3; j++)  
        System.out.println("Ha!");
```

- (a) 3 (b) 4
- (c) 40 (d) 41
- (e) 43 (f) 44
- (g) 120 (h) 123

5. (2pts) What is the result of running the following code?

```
int x = 33;
int y = 34;
if (++x == y)
    System.out.println("x: " + x + "y: " + y--);
else
    System.out.println("y: " + y-- + "x: " + x);
```

Program Output:

A) x: 34 y: 34 B) y: 34 x: 34
C) x: 33 y: 34 D) y: 34 x: 33
E) x: 34 y: 33 F) y: 33 x: 34
G) x: 33 y: 33 H) y: 33 x: 33

6. (2pts) Which of the following code fragment prints out all the numbers from [100,0] that are both multiples of 7 and multiples of 2?

(a)

```
int i = 100;
while (i == 0) {
    if ( i%7==0 && i%2==0 )
        System.out.println(i);
    i--;
}
```

(b)

```
int i = 100;
while (i >= 0) {
    if ( i%7==0 && i%2==0 )
        System.out.println(i);
    i--;
}
```

(c)

```
int i = 100;
while(i == 0 && i%7==0 && i%2==0) {
    System.out.println(i);
    i--;
}
```

(d)

```
int i = 100;
while(i >= 100 && i%7==0 && i%2==0) {
    System.out.println(i);
    i--;
}
```

(e)

```
int i = 100;
while (i == 0) {
    if ( i%7==0 || i%2 ==0 )
        System.out.println(i);
    i--;
}
```

(f)

```
int i = 100;
while (i >= 0) {
    if ( i%7==0 || i%2==0 )
        System.out.println(i);
    i--;
}
```

(g)

```
int i = 100;
while(i == 0 || i%7==0 || i%2==0) {
    System.out.println(i);
    i--;
}
```

(h)

```
int i = 100;
while(i >= 0 || i%7==0 || i%2==0) {
    System.out.println(i);
    i--;
}
```

(6pts total) The following code will be used to answer the next few questions. **READ IT CAREFULLY.**

```
import java.util.Scanner;

public class Test
{
    public static void main(String [] args)
    {
        Scanner in = new Scanner(System.in);
        String foo = in.nextLine();

        if (foo.length() % 2 ==0)
        {
            if (foo.indexOf("!") >=0)
            {
                System.out.print("Hurray ");
            }
            else if (foo.charAt(foo.length()-1) == '?')
            {
                System.out.print("What ");
            }
            System.out.print("Oh, yeah");
        }
        else
        {
            int i = foo.indexOf("*");
            int j = foo.indexOf("*", i+1);
            System.out.println(foo.substring(i,j));
        }
    }
}
```

7. (2pts): What does this program do when the user inputs:

Java is Fun

- | | |
|--------------------|------------------|
| A) Hurray Oh, yeah | B) Oh, yeah |
| C) What | D) What Oh, yeah |
| E) *Java* | F) *Java |
| G) is Fun | H) * is Fun |

8. (2pts): What does this program do when the user inputs:

Rocks?

- | | |
|--------------------|------------------|
| A) Hurray Oh, yeah | B) Oh, yeah |
| C) What | D) What Oh, yeah |
| E) *Java* | F) *Java |
| G) is Fun | H) * is Fun |

Page 3

MIDTERM

REAL NAME: _____

CSE8A

PID: _____

9 (2pts): Which of the following statements MOST ACCURATELY AND PRECISELY describes the conditions under which “Hurray” is printed?

- A) The length of the entered String is **even** and the String ends with an exclamation point
- B) The length of the entered String is **even** and the String does not end with an exclamation point
- C) The length of the entered String is **even** and the String contains an exclamation point
- D) The length of the entered String is **even** and the String does NOT contain an exclamation point
- E) The length of the entered String is **odd** and the String ends with an exclamation point
- F) The length of the entered String is **odd** and the String does not end with an exclamation point
- G) The length of the entered String is **odd** and the String contains an exclamation point
- H) The length of the entered String is **odd** and the String does NOT contain an exclamation point

10 (2pts). Which of the following codes correctly implement a validation loop to ensure that the user enters a number between 1 and 9 inclusive? **MORE THAN ONE ANSWER MAY BE CORRECT.**

Assume this line has already been executed: `Scanner in = new Scanner(System.in);`

CIRCLE ALL CORRECT ANSWERS FOR FULL POINTS!

A)

```
int ans;
ans = in.nextInt();
while (ans > 0 && ans < 10)
{
    ans = in.nextInt();
}
```

B)

```
int ans;
ans = in.nextInt();
while (ans > 0 || ans < 10)
{
    ans = in.nextInt();
}
```

C)

```
int ans;
ans = in.nextInt();
while (ans <= 0 && ans >= 10)
{
    ans = in.nextInt();
}
```

D)

```
int ans;
ans = in.nextInt();
while (ans <= 0 || ans >= 10)
{
    ans = in.nextInt();
}
```

E)

```
int ans;
do
{
    ans = in.nextInt();
} while(ans <= 0 && ans >= 10);
```

F)

```
int ans;
do
{
    ans = in.nextInt();
} while(ans <= 0 || ans >= 10);
```

WRITE YOUR ANSWERS ON THIS PAGE

11. (3 pts) If you want to convert Celsius to Fahrenheit, the *mathematical* equation is

$$F = \frac{9}{5} C + 32$$

Below, write a short, **legal, Java code segment** to declare appropriate variables (with good style, naming convention, and appropriate type), read in the Celsius value from the user, and calculate and store the value in a variable.

12. (2pts) What is the output produced when the following code is executed?

```
String foo = "Java!";
String bar = "";

for (int i = 0; i < foo.length(); i+=2)
{
    bar = foo.charAt(i) + bar;
}
System.out.println(bar);
```

Output: _____

13. (2pts) Write a single line of legal Java code to declare an array of 25 `ints` named `foo`.

Answer: _____

WRITE YOUR ANSWER ON THIS PAGE

14. (3pts) Explain what the following code does in plain English sentences. State your answer **SUCCINCTLY** telling us at a **HIGH LEVEL** what the code does overall.

```
public static void main(String[] args)
{
    double x, y;
    Scanner in = new Scanner(System.in);

    x = in.nextDouble();
    y = -1.0;

    while ( x >= 0.0 )
    {
        if ( y < x )
            y = x;
        x = in.nextDouble();
    }

    System.out.println(y);
}
```

WRITE YOUR ANSWER ON THIS PAGE

15. (8pts) Write a program which reads in values from the user until they enter the **sentinel** -999.

- The program should calculate and output the average of all the even numbers entered.
- The program should also count and output how many of the values entered by the user were negative (not including the sentinel value).

Style, conciseness, and appropriate use of programming constructs will be graded in conjunction with correctness.

```
import java.util.Scanner;
public class UserTest
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
```

```
    }
}
```

Page 7

MIDTERM

REAL NAME: _____

CSE8A

PID: _____

USEFUL STRING CLASS METHODS (from pages 38-41 in the text)

int length()

Returns the length of the calling object (which is a string) as a value of type int.

int indexOf(A_String)

Returns the index (position) of the first occurrence of the string `A_String` in the calling object string. Positions are counted 0,1,2, etc. Returns -1 if `A_String` is not found.

int indexOf(A_String, Start)

Returns the index (position) of the first occurrence of the string `A_String` in the calling object string that occurs at or after position `Start`. Positions are counted 0,1,2, etc. Returns -1 if `A_String` is not found.

char charAt(Position)

Returns the character in the calling object string at the `Position`. Positions are counted 0,1,2, etc.

String substring(Start,End)

Returns the substring of the calling object string starting from position `Start` through, but not including, position `End` of the calling object. Positions are counted 0,1,2, etc.

MIDTERM

REAL NAME: _____

CSE8A

PID: _____