

S1. Multiple Choice (1 point each):

- 1) What will be the value of the following expression?

 $\text{Math.pow}(3, 2 \% 1) * (1 / 5 + 4 * 4 - 2 / 6 * 3)$

- a) 0.0
- b) 15.2
- c) 16.0
- d) 136.8
- e) None of the above

Use the code segment below for questions 2 – 9. This code should have two arrays for storing names and ages for people. The arrays will be filled by reading from a file called “people.log”. We can assume that the number of people in the file is not larger than 100. Fill in the blanks of the given code for questions 2 – 8.

```
__String[]__ name = new __String[]__;  
2)
```

```
__int[]__ age = __int[]__;  
3)
```

```
__File__ file = new __File__("people.log");  
4)
```

```
__Scanner__ input = new __Scanner__(__file__);  
5) 6)
```

```
for(int i = 0; input.hasNext(); i++){  
7)
```

```
    name[i] = input.__next();  
8)
```

```
    age[i] = input.nextInt();
```

```
}
```

- 9) The code above will not compile in a typical method (main or otherwise). We must include code to handle an error if the file is not found. We can do this through which **two** of the following?
- a) Changing the file name to a ".txt" file
 - b) Including “throws Exception” as part of the method signature
 - c) Storing ages as Strings instead of integers
 - d) Using a try-catch block
 - e) Giving up on Computer Engineering and choosing a different direction in life

- 10) What will be the value of the following expression?

 $16 / 2 * 3 + 4 * 2 + 13 / 5$

- a) 13.8
- b) 29
- c) 34
- d) 34.0
- e) 34.6

- 11) If we want to display the first five (5) characters of a String called longString, which of the following will do this?

a) `System.out.print(longString.substring(0, 5));`
b) `System.out.print(longString.substring(0, 6));`
c) `System.out.print(longString.substring(1, 5));`
d) `System.out.print(longString.substring(1, 6));`
e) `System.out.print(longString.substring(5));`

Use the code segment below for questions 12 – 19. This code should return true if the number passed is odd and false if it is even. Fill in the blanks of the given code for questions 12 – 15.

```
public static ____ boolean ____ isOdd(int myVar){  
    12)  
  
    ____ return ____ myVar ____ % 2 ____ != 0;  
    13)          14)          15)  
}
```

- 16) In the code segment, isOdd is a(n) ____.

a) **method**
b) loop
c) variable
d) iterator

- 17) In the code segment, myVar is a(n) ____.

a) iterator
b) **parameter**
c) operator
d) method

- 18) Which of the following is a **valid** way to use isOdd in an **assignment** expression?

a) `isOdd(5) == isOdd(6)`
b) `boolean odd = 7.isOdd();`
c) **`int remainder = (isOdd(4) ? 1 : 0);`**
d) `isOdd(8) = 0;`
e) `isOdd(9) > isOdd(10)`

- 19) Which of the following expressions evaluates to **False** given the following initial conditions?

`int zero = 0, one = 1, two = 2, three = 3;`
a) `(two == 3) == (three == 2)`
b) `(one = 2) == (two = 2)`
c) `(zero == 0) == (one == 1)`
d) **`(zero = 3) == (three = 0)`**
e) None of these evaluates to False

- 20) All of the following are part of a recursive method definition except:

a) **Recursive Call**
b) **Convergence**
c) **Base Case**
d) Decrementing a variable (i.e. "n - 1")
e) All of the above are part of the recursive definition

Use the following loop to answer questions 21 – 23:

```
String s = "I have a bad feeling about this";  
int i = s.length() - 1;  
while(s.charAt(i) != 'i' && i > 0)  
    System.out.println(s.charAt(i--));
```

21) What will be the value of i after the loop is completed executing?

- a) 0
- b) 17
- c) 29
- d) 30
- e) Cannot be determined

22) How many total times will body of the loop execute?

- a) 0
- b) 1
- c) 2
- d) 30
- e) 31
- f) Infinite loop

23) What will be the final value displayed?

- a) 's'
- b) 'i'
- c) 'n'
- d) 'I'
- e) A space character

24) If we want to display a String called myString but with all spaces removed, which of the following will do this?

- a) `System.out.print(myString.removeSpaces());`
- b) `System.out.print(myString.toLowerCase());`
- c) `System.out.print(myString.replace(" ", ""));`
- d) `System.out.print(myString.find(" "));`
- e) `System.out.print(myString.indexOf(" "));`

Use the following loop to answer questions 25 and 26:

```
int i = 0;  
while (i < 2019){  
    System.out.println(i);  
    i += 1;  
}
```

25) What will be the value of i after the loop is completed executing?

- a) 0
- b) 2018
- c) 2019
- d) 2020
- e) Cannot be determined

26) How many total times will the loop execute?

- a) 0
- b) 2018
- c) 2019
- d) 2020
- e) Infinite loop

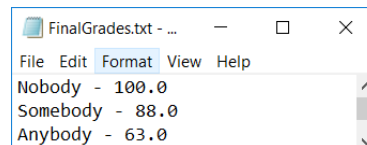
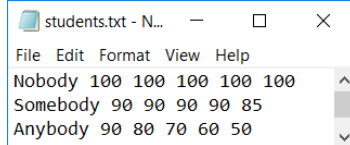
True/False (1 point each):

- 27) **T F** A recursive solution for a problem is never as good as an iterative one.
- 28) **T F** Given an `int` variable `intVar`:
`intVar < 0 && intVar == 0 && intVar > 0`
- 29) **T F** Two methods in a class can have the same name.
- 30) **T F** The following two lines of code do the same thing:
`myVar -= 3;`
`myVar = 3 - myVar;`
- 31) **T F** The number of parameters, the order of parameters AND the type of parameters in a method call must match the argument list in the method definition. Otherwise, the method call will result in an error.

A	B	C	$A !A$	$A \&\& !B !C$
False	False	False	31) T	
False	False	True		
False	True	False		32) T
False	True	True		
True	False	False		
True	False	True		
True	True	False	33) T	34) T
True	True	True		

S2. Find and Fix the Error (15 points)

We want to read in a file (students.txt) that has students with five scores (these are lab activity average, quiz average, homework average, midterm exam, final exam...A screen shot of an example file is shown). For each student, we pass the values to a method that calculates the final grade using an array of weights for each value (10% for activities and quizzes, 20 % for homework and midterm, 40% for final). We return the calculated average and use it along with the student's name to output to a second file (FinalGrades.txt) on each line the student's name and calculated average (Example expected output shown below).



Unfortunately, in the code we wrote for this (shown), we experienced some errors. Each time we fixed an error, another was encountered. In fact, we found five (5) errors. For each error do the following:

1. Circle the error in the original code (1 point)
2. State the type of error (Compiler, Run-time, Logical) (1 point)
3. Rewrite the code corrected (1 point)

You may assume any statements not shown (import, public class, etc.) are correct and do not have errors. You may also assume there are no errors in the source text file.

```
public static void main(String[] args) throws Exception {
    Scanner input = new Scanner(new File("students.txt"));
    PrintWriter output = new PrintWriter(new File("FinalGrades.txt"));
    String name;
    double[] scores = new double[5];
    double[] weights = {0.1, 0.1, 0.2, 0.2, 0.4};
    while(input.hasNext()); {
        name = input.next();
        for(int i = 0; i < 5; i++)
            scores[i] = input.nextDouble();
        System.out.println(name + " - " + calcFinalGrade(scores, weights));
    }

    public static double calcFinalGrade(double[] scores, double[] weights) {
        double finalGrade = 0;
        for(int i = 0; i < scores.length; i++)
            finalGrade += scores[i] * weights[i];
        System.out.println(finalGrade);
    }
}
```

Hints:

1. The first error message was
"error: illegal start of expression public static
double calcFinalGrade(double[] scores, double[]
weights) {"
2. Then, we received an error message of
"error: missing return"
3. When we fixed these two, we compiled
but when we ran the code, it never stopped.
4. We fixed that, but when we run the code,
the correct information is displayed to the screen
instead of the file.
5. We corrected that, but then the file was
empty.

S3. What is the Output? (8 points each)

- 1) Determine the output produced by the following code:

```
String s = "CSE 101 is BITTI";
for(int i = 0; i < s.length(); i++)
    if(s.charAt(i) >= '9' && s.charAt(i) <= '1')
        System.out.print("102");
    else if (s.charAt(i) == 'I') {
        System.out.print("SONRA");
        break;
    } else {
        System.out.print(s.charAt(i));
        i++;
    }
System.out.print("\n");
```

CE11i SONRA

- 2) Determine the output produced by the following code:

```
int[][] arr = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
double total = 0;
for (int i = 1; i < 3; i++){
    for(int j = 0; j < 2; j++)
        if (arr[i][j] % 2 == 0)
            total -= arr[i][j] * 1.5;
        else
            total += arr[i][j];
    System.out.println(total);
}
```

-1.0
-6.0

S4. Short Program Fragment (8 points each)

In the box provided, write a **recursive** method called *addDigits(int n)* that returns an integer that is the sum of the digits of *n*. Examples shown. Identify the base case, recursive call, and convergence for your solution.

addDigits(1234) -> returns 10

addDigits(20200106) -> returns 11

```
public static int addDigits(int n){
```

Consider the recursive method to the right.

In the box provided, write the result of calling *mystery(5, 9)* and the result of calling *mystery(2, 25)*. Give a general definition for what *mystery(int a, int b)* calculates.

```
public static int mystery(int a, int b) {  
    if (b == 0)  
        return 0;  
    if (b % 2 == 0)  
        return mystery(a+a, b/2);  
    return mystery(a+a, b/2) + a;  
}
```

S5. Develop a Solution (18 points)

We want to create a method that produces an integer array of range of values based on some given inputs. Create the following three versions of the `range()` method. It takes the given parameters and returns the desired array.

`int[] range(int start, int end, int step)` produces an array of values starting at *start* up to but not including *end* with the difference between each number in the array being *step*

`int[] range(int start, int end)` produces an array of values starting at *start* up to but not including *end* stepping by 1

`int[] range(int end)` produces an array of values starting at 0 up to but not including *end* stepping by 1

<code>range(5, 15, 3) -> [5, 8, 11, 14]</code>
<code>range(5, 10) -> [5, 6, 7, 8, 9]</code>
<code>range(10) -> [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]</code>

Examples ->

Part 1 – Identify Subproblems (3 points)

Identify at least 3 subproblems for this problem.

1. _____
2. _____
3. _____

Part 2 – Make a Plan (6 points)

Make a plan for how you will solve this problem (and sub-problems) using any combination of writing, pseudocode, or flowchart.

Part 3 – Java Code (6 points)

Write the Java code for implementing your solution to this problem.

Part 4 – Example (3 points)

What does your method return when you call it with `range(0)`?