

1. Use a decision structure to write an appropriate statement for each of the following
  - a) **Display Great job! When the grade is 90 or higher.**

```
if (grade >= 90)
    System.out.println("Great job!");
```

- b) **Display error when number is less than 20 or greater than 50**

```
if (number < 20)
    System.out.println("Error");

if (number > 50)
    System.out.println("Error");
```

- c) **Add 2 to the value of y when y is less than 100**

```
if (y < 100)
    y = y + 2;
```

2. Assume num1 and num2 contain integer values. Write an if-else if statement that displays one of the following messages as appropriate

```
if (num1 > num2)
    System.out.println("num1 is greater than num2");
else if (num1 < num2)
    System.out.println("num1 is less than num2");
else
    System.out.println("num1 is equal to num2");
```

3. a) **which is the appropriate word, odd or even for the blanks below?**

```
If (num % 2==0) {
    System.out.println ("even number");
} else {
    System.out.println("odd number");
}
```

**b) Rewrite the if-else as a switch statement**

```
switch (num % 2) {  
    case 0:  
        System.out.println("even number");  
        break;  
    case 1:  
        System.out.println("odd number");  
        break;  
}
```

4. Write statements that use Math.random() to generate random numbers for each of the following situations

- a) Generate a random integer between 1 and 50

```
int number = (int)(Math.random() * 50) + 1;  
System.out.println(number);
```

- b) Generate a random integer between 20 and 100

```
int number = (int)(Math.random() * 81) + 20;  
System.out.println(number);
```

- c) Generate a random double between 10 and 20, inclusive

```
double number = (Math.random() * 10) + 10;  
System.out.println(number);
```

5. Identify the logic errors in the statement below, which should display a single appropriate message for any value of age

```
if (age < 18) {  
    System.out.println("child");  
} else if (age > 18 && age <65) {  
    System.out.println("adult");  
} else if age (age > 65) {  
    System.out.println("senior");  
}
```

There are multiple syntactic mistakes in the code. Initially, the curly quotes that surround the strings, like "child," are not supported by Java; instead, "child" or other straight quotes should be used. Furthermore, the last else if statement's grammar is flawed; instead of writing else if age (age > 65), it should read else if (age > 65). Lastly, age <65 has a small spacing problem; it is cleaner to write age < 65.

The criteria contain logical problems in addition to syntax mistakes. Age 18 is not covered by either of the conditions: age < 18 covers ages 0–17, while age > 18 & < 65 covers ages 19–64, while age 18 is not included in either condition. In a similar vein, 65 is

Fixed Code:

```
if (age < 18) {  
    System.out.println("child");  
}  
} else if (age < 65) { // If it gets here, age is >= 18. We only need to check if it's < 65.  
    System.out.println("adult");  
}  
} else { // If it gets here, age is not < 18 and not < 65, meaning age is >= 65  
    System.out.println("senior");  
}
```

6. Given the following assignments, determine if each of the following expressions evaluates to true or false

size=100 weight =50 value =75

- a) Size > 50 && weight ==50 - true
- b) Value < 100 && !(weight == 500) -false
- c) size>= 100 || value >= 100 -true
- d) Weight <50 || size>50 -true
- e) !(value < 75) -true
- f) !(size > 100 && weight >50 && value > 75) -true
- g) (value <125 || weigh <76) && size == 100 -true

8. Determine if each of the following are true or false. If false, explain why.

- a) The condition of an if statement must be a boolean expression- true
- b) A nested if statement and an if-else if statement are the same- **false because a nested if does is not the same as if else if**
- c) The expression in a switch statement must evaluate to a double- **false because switch does not allow double to be used**
- d) Numbers generated by a computer program are actually pseudorandom-true
- e) The (double) cast is needed to generate a random integer- **false you don't need double to cast a random int**
- f) A compound boolean expression can contain more than two boolean expressions-true
- g) In a logical and expression, both operands must be true for the expressions to evaluate to true -true
- h) In logical expressions, && is evaluated before ! -**false because ! is evaluated before &&**
- i) The pow() method in the math class is used for exponentiation-true
- j) The statement x=abs(-3); will return the value of 3-**false because it should be Math.abs(-3)**