1. Decision Structures

a)

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

b)

```
if (number < 20 || number > 50) {
    System.out.println("Error");
}
```

c)

```
if (y < 100) {
    y = y + 2;  // or y += 2;
}
```

2. If-Else If Statement

```
if (num1 > num2) {
    System.out.println("First number is larger.");
} else if (num2 > num1) {
    System.out.println("Second number is larger.");
} else {
    System.out.println("Numbers are equal.");
}
```

3. Odd or Even

a) Fill in the blanks:

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

b) Rewrite using switch:

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

4. Random Numbers

Assume we imported:

```
import java.util.Random;
Random rand = new Random();
```

a) Between 1 and 50:

```
int r = rand.nextInt(50) + 1;
// gives 1–50 inclusive
```

b) Between 20 and 100:

```
int r = rand.nextInt(81) + 20;
// (100 - 20 + 1 = 81)
```

c) Random double between 10 and 20:

```
double r = 10 + (rand.nextDouble() * 10);
// 10.0 to 20.0 inclusive
```

5. Logic Errors in Age Example

Original:

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

Problems:

- Age 18 does not fit any category (missed case).
- Age 65 also doesn't fit any category (should be senior).

Corrected Version:

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

6. True or False

a) true
b) false
c) true
d) true
e) true
f) true
g) true

7. True or False

a) The condition of an if statement must be a Boolean expression.
True – In Java (and most languages like C, C++), the condition in an if must evaluate to a Boolean (true or false).

b) A roundoff error can occur when comparing two integers.
False – Integers are stored exactly in binary (no rounding), so no roundoff error occurs. Roundoff errors happen with floating-point numbers.

c) A nested if statement and an if-else if statement are the same.
False –

Nested if means one if inside another.

if-else if is sequential checking of conditions.
They are structured differently.

d) The expression in a switch statement must evaluate to a double.
False – switch expressions can be byte, short, char, int, String, or enum types. Not double.

e) Numbers generated by a computer program are actually pseudorandom.
True – Computers use deterministic algorithms for "random" numbers, so they are pseudorandom.

f) Specifying a seed value results in a different sequence of "random" numbers each time the program is run.
False – If you specify the same seed, you'll always get the same sequence of numbers. To get different sequences, you don't specify a fixed seed (or use something like the current time).

g) A compound Boolean expression can contain more than two Boolean expressions.
True – You can chain Boolean expressions with &&, ||, etc. (e.g., (x > 5 && y < 10 || z == 3)).

h) In a logical AND expression, both operands must be true for the expression to evaluate to true.
True – That's the definition of &&.

i) In logical expressions, && is evaluated before ||.
True – Operator precedence rules: && has higher precedence than ||.

j) The pow() method in the Math class is used for exponentiation.
True – Math.pow(base, exponent) performs exponentiation in Java.

k) The statement x = abs(-3); will return the value 3.
False – Java's absolute value function is Math.abs(), not abs(). As written, abs(-3) would be an error unless abs is user-defined.

l) A diamond shaped object represents a decision in a flowchart.
True – In flowchart notation, a diamond represents a decision (yes/no or true/false).