Question 1-B

Switch statement are not supported floating-point types like “float” and “double”

Question 2-B

Since 6>meal is false so – – tip is evaluated. As a result, the type variable 1 is reduced.

Question 3-C

Since this objects are not the same, the == test on them evaluates to false. The equals() test on them returns true because the values they refer to are equivalent.

Question 4-D

Syntax error on token "else", this token must delete. The second if statement is not connected to the last else.

Question 5-C

Default statement doesn't take a value, unlike a case statement.

Question 6-B

long thatNumber = 5 >= 5 ? 1+2 : 1*1; // thatnumber=3

if(++thatNumber < 4) // thatnumber= 4

thatNumber += 1;

Question 7-B

The break statement exits a switch statement, skipping all remaining branches

Question 8- C

Ternary expressions are commonly used to replace short if-then-else statements

Question 9-C

CandidateA and candidateB are numbers, but the && operation can only be applied to boolean expressions.

Question 10-A

The İf statement 6 % 3 evaluates to 0, since there is no remainder, and since 0 >= 1 is false, the expression triceratops++ is not called. – – Triceratops is executed, resulting in a value of 2

Question 11-D

if-then statement may execute a single statement or a block of code {}.

Question 12-D

The output of the code is “Not enough Too many”

Question 13-B

A case statement can end with a break statement, but it isn’t compulsory

Question 14-D

Boolean expression && or || corresponds to the truth table that only evaluates to true if both operands are true.Only the conjunctive logical && operator represents this relationship,

Question 15-C

Java does'nt automatically convert integers to boolean values for use in if-then statements.

Question 16-B

The pre-increment [++v] operator increases the value of a variable by 1 and returns the new value

The post-decrement [v--] operator decreases the value of a variable by 1 and returns the original value

Question 17-B

int tiger = 2;

short lion = 3;

long winner = lion+2*(tiger + lion); // winner = 3+2*(3+2)=13

Question 18-B

Since switch statements do not support long values, and long

cannot be converted to int without a possible loss of data

Question 19-D

The day value is an int, not a boolean expression, in the second ternary operation.

Question 20-C

Leaders variable is undefined. There are missing brackets.

Question 21-B

System.out.print(5 + 6 + "7" + 8 + 9); //11717

Question 22-B

The subtraction - operator is used to find the difference between two numbers.

The modulus % operator is used to find the remainder when one number is divided by another

Question 23-C

int dog = 11; int cat = 3;

int partA = dog / cat; // 3

int partB = dog % cat; // 2

int newDog = partB + partA * cat; // 2+3*3=11

Question 24-B

int flavors = 30; int eaten = 0;

switch(flavors) {

case 30: eaten++; //1

case 40: eaten+=2; //3

default: eaten--; // 2

}

System.out.print(eaten); // **2**

Question 25-C

On line 4. The code doesn't compile. Because of t ype mismatch. Can not convert from int to String

Question 26-A

If they are the same String object, equals() will trivially return true.

Question 27 –B

myTestVariable.equals(null) = False

Since we are given that myTestVariable is not null, the statement will evaluate to false

Question 28-D

On line 8. (streets && intersections > 1000) is invalid because streets is not a boolean expression and cannot be used as the left-hand side of the conjunctive logical && operator

Question 29-D

On line The & operator always evaluates both operands, while the && operator may only evaluate the right operand.

Question 30-C

int x = 10, y = 5;

boolean w = true, z = false;

```
x = w ? y++ : y--;                                    // y=6
w = !z;                                               // true
System.out.print((x+y)+" "+(w ? 5 : 10));             //   (6+5)   5
```

Question 31-A

```
String bob = new String("bob");
String notBob = bob;
System.out.print((bob==notBob)+" "+(bob.equals(notBob)));        //  true   true
```

Question 32-B

in java, There are parentheses and operator precedence

Question 33-D

The XOR ^ operator evaluates to true if p and q differ and false if they are the same.

Question 34-?

Question 35-C

/, *, %

This three operators have the same order of precedence.

Question 36-D

The ^ operator can only be applied to boolean values

Question 37-C

x || y   corresponding to when one of them is true

Question 38-D *

Since red is missing the final attribute, no variable type allows the code to compile

Question 39-C

5.21<= X <8.1

Question 40-B

```
int turtle = 10 * (2 + (3 + 2) / 5);                                  //30
int hare = turtle < 5 ? 10 : 25;                                      //25
System.out.print(turtle < hare ? "Hare wins!" : "Turtle wins!");      // Turtle wins
```

Question 41-A

All of the terms of getResult() in this question evaluate to 0, since they are all less than or equal to 5. Therefore result 0+0+0+0+"".

Question 42-?

Question 43-D

(OR) || operator is true if either of the operands are true

The logical complement (!) operator reverses or flips a boolean value

Question 44-A

int characters = 5; int story = 3;

double movieRating = characters <= 4 ? 3 : story>1 ? 2 : 1;   // 2

Question 45-B

A switch statement can have any number of case statements and at most one default statements.

Question 46-A *

Question 47-D

expression is not compiled and executed. !2 there is no such use.

Question 48-?

Question 49-A

Sorting should be this way: - ,   + ,   / ,   *

Question 50-C

The code doesn't compile due to p1. Type mismatch: cannot convert from int to String