TYPE:B-DATA HANDLING:CH-7

1) What is the result produced by (i) bool (0) (ii) bool (str(0))? Justify the outcome.

sol:

(i) False
(ii)True

2) What will be the output, if input for both the statements is 5 + 4/2.

```
6 == input ("Value 1:")
6 == int(input ("value 2:"))
```

sol:

Output of first statement is False as '5 + 4/2' is entered as a string so it cannot be equal to the number 6.

The second statement gives an error as int() function cannot convert the string '5 + 4/2' to a valid integer.

3) Following Python code has an expression with all integer values. Why is the result in floating point form?

```
a, b, c = 2, 3, 6
d = a + b * c/b
print(d)
```

sol:

in python the division operator always reture the value in the floating points and when the floating point is multiplies with anyother number the result must be in the floating point value hence the above expression results in the floating points

4)(a) What will following code print?

```
a = va = 3
b = va = 3
print (a, b)
```

sol:

3 3

4)(b) What will following code print?

```
a = 3
b = 3.0
print (a == b)
print (a is b)
```

```
sol:
  True
  False
  3 3.0
```

5(a) What will be output produced by following code? State reason for this output.

```
a, b, c = 1, 1, 2
d = a + b
e = 1.0
f = 1.0
g = 2.0
h = e + f
print(c == d)
print(c is d)
print(g == h)
print(g is h)

    sol:
      True
      True
      True
      False
```

5(b) What will be output produced by following code? State reason for this output.

```
a = 5 - 4 - 3
b = 3**2**3
print(a)
print(b)

    sol:
      -2
      6561
```

5(c) What will be output produced by following code? State reason for this output.

```
a, b, c = 1, 1, 1
d = 0.3
e=a+b+c-d
f=a+b+c == d
print(e)
print(f)

    sol:
      2.7
      False
      1
      1
```

6) Make change in the expression for z of previous question so that the output produced is 0. you cannot change the operators and order of variables (Hint: use a function around a sub-expression)

sol:

```
x, y = 4, 8
z = int(x/y)*y
print(z)
```

7) Consider the following expression. X= ‘and’ * (3+2) > ‘or’ + ‘4’ what is the data type of the value that is computed by the expression.

sol:

The final result of the expression is a boolean value (False), so the data type is bool.

8)(a) What will be the output of following Python code?

```
a = 12
b = 7.4
c = 1
a -= b
print(a, b)
a *= 2 + c
print(a)
b += a * c
print(b)
```

sol:

```
4.6 7.4
13.799999999999999
21.2
```

8(b) What will be the output of following Python code?

```
x, y = 4, 8
z = x/y*y
print(z)
```

sol:

```
4.0
```

9) Consider the following code segment:

```
a = input()
b = int(input())
c = a + b
print(c)
```

sol:

error rises as a is a string and b is a integer thus they both cannot be added

```
a = int(input())
b = int(input())
c = a + b
print(c)
```

10) Consider the following code segment:

```
a = input("Enter the value of a:")
b = input("Enter the value of b:")
print(a + b)
```

If the user runs the program and enters 11 for a and 9 for b then what will the above code display?

sol:

```
119
```

11) Find out the error and the reason for the error in the following code. Also, give the corrected code.

```
a, b = "5.0", "10.0"
x = float(a/b)
print(x)
```

sol:

firstly the error occurs while trying to operate the division operation with two strings a and b
secondly the error occurs when try to convert the divide string as float value

12) Consider the following program. It is supposed to compute the hypotenuse of a right triangle after the user enters the lengths of the other two sides.

```
a = float(input("Enter the length of the first side:"))
b = float(input("Enter the length of the second side:"))
h = sqrt(a * a + b * b)
print("The length of the hypotenuse is", h)
```

When this program is run, the following output is generated (note that input entered by the user is shown in bold):

sol:

this error happened as sqrt() is a function used from the math module and module is not imported here it rises an error . importing module will be explained clearly is chapter 8

13) After adding import math statement, we need to change the line h = sqrt(a * a + b * b) to h = math.sqrt(a * a + b * b). The corrected working code is below:

```
a = float(input("Enter the length of the first side:"))
b = float(input("Enter the length of the second side:"))
h = math.sqrt(a * a + b * b)
print("The length of the hypotenuse is", h)
```

14) Which of the following expressions will result in an error message being displayed when a program containing it is run?

(a) 2.0/4

(b) "3" + "Hello"

(c) 4 % 15

(d) int("5")/float("3")

(e) float("6"/"2")

sol:

(a) No Error

(b) No Error

(c) No Error

(d) No Error

(e) This will cause an error of unsupported operand types as using division operator on string types is not allowed in Python.

15) Following expression does not report an error even if it has a sub-expression with 'divide by zero' problem:
3 or 10/0

sol:

Because the division by zero never actually occurs during the execution, no error is reported [1].

16) What is the output produced by following code?

```
a, b = bool(0), bool(0.0)
c, d = str(0), str(0.0)
print (len(a), len(b))
print (len(c), len(d))
```

sol:

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 3, in <module>
TypeError: object of type 'bool' has no len()
```

17) Predict the output if e is given input as 'True':

```
a = True
```

```
b = 0 < 5
print (a == b)
print (a is b)
c = str (a)
d = str (b)
print (c == d)
print (c is d)
e = input ("Enter :")
print (c == e)
print (c is e)
```

sol:

```
True
True
True
True
Enter :True
True
False
```

18)(a) Find the errors(s)

```
name = "HariT"
print (name)
name[2] = 'R'
print (name)
```

sol:

this program trying to change the 2nd index of thr given string. but as the string is immutable this code rises error.

18)(b) Find the errors(s)

```
a = bool (0)
b = bool (1)
print (a == false)
print (b == true)
```

sol:

```
false
false
```

18)(c) Find the errors(s)

```
print (type (int("123")))
print (type (int("Hello")))
print (type (str("123.0")))
```

sol:

```
<class 'int'>
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 2, in <module>
ValueError: invalid literal for int() with base 10: 'Hello'
<class 'str'>
```

18)(d) Find the errors(s)

```
pi = 3.14
print (type (pi))
print (type ("3.14"))
print (type (float ("3.14")))
print (type (float("three point fourteen")))
```

sol:

the error rises at the last statement , as that statement trying to convert the string as floating point which is not possible.

18)(e) Find the errors(s)

```
print ("Hello" + 2)
print ("Hello" + "2")
print ("Hello" * 2)
```

sol:

only the first statement rises error, the rule of ‘+’ operator is both the operand must be of same data type, but here they give two different data type.

18)(f) Find the errors(s)

```
print ("Hello"/2)
print ("Hello" / 2)
```

sol:

the both the statements rises an error because we know that the division operator cannot be used with strings. it can be used only with the integers and floating points.

20)(a) What will be the output produced?

```
x, y, z = True, False, False
a = x or (y and z)
b = (x or y) and z
print(a, b)
```

sol:

```
True False
```

20)(b) What will be the output produced?

```
x, y = '5', 2
z = x + y
print(z)
```

sol:

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 4, in <module>
TypeError: can only concatenate str (not "int") to str
```

20)(c) What will be the output produced?

```
s = 'Sipo'
s1 = s + '2'
s2 = s * 2
print(s1)
print(s2)
```

sol:

```
Sipo2
SipoSipo
```

20(d) What will be the output produced?

```
w, x, y, z = True , 4, -6, 2
result = -(x + z) < y or x ** z < 10
print(result)
```

sol:

```
False
```

21) Program is giving a weird result of "0.50.50.50.50.50.50..........". Correct it so that it produces the correct result which is the probability value (input as 0.5) times 150.

```
probability = input("Type a number between 0 and 1: ")
print("Out of 150 tries, the odds are that only", (probability * 150), "will succeed.")
```

[Hint. Consider its datatype.]

sol:

The program produces the incorrect result because `input()` returns a string, not a number. When the string `"0.5"` is multiplied by `150`, Python repeats the string instead of performing a mathematical calculation, resulting in `"0.50.50.50..."`. To fix this issue, the input must be converted to a numeric data type such as `float` before performing the multiplication. After converting the input to a float, the program correctly calculates the probability multiplied by 150.