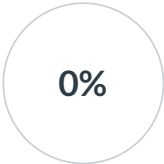


Results

Shang Xiao — 2nd Attempt

5 questions require grading



0
Out of 10 points

26:41
Time for this attempt

This assessment has unlimited attempts. Take Now

Attempt History

Results	Points	Score	(Highest score is kept)
Attempt 1	9 of 10	90%	(Highest score)
Attempt 2	0 of 10	0%	

Your Answers:

1 2 points possible

What are 2 variable types available in python?

Numbers: Whole numbers, aka intergers (int) and decimals (float)

String: using with " ", or ', such as "this is a strong", things inside " " are known as strings

Waiting for grade

2 2 points possible

Describe what the operators % and not do. Give a line of code for each of them that shows how they are used.

% is a remainder operator, when used, it gives out the results of the remainder of the calculation.

For example, if we use $5\%2$, called 5 mod 2, then it will return us 1, because 5 divide by 2 is 2 with a remainder of 1, and that is why

An % operator is different from the // operator, as the // operator only gives us the integer division result, and returns us the division

For example, $11\%3=2$ because $11/3=3$ remainder 2, and that is why we get 2.

For example, $11//3=3$ because $11/3=3$ remainder 2, and that is why we get 3.

not is a logical (Boolean) operator and is unitary. It simply means negation, it negates everything afterwards and reverses the value (a<b) is equalvant to $a\geq b$, as "not" reverses the value (a<b), which is $a\geq b$.

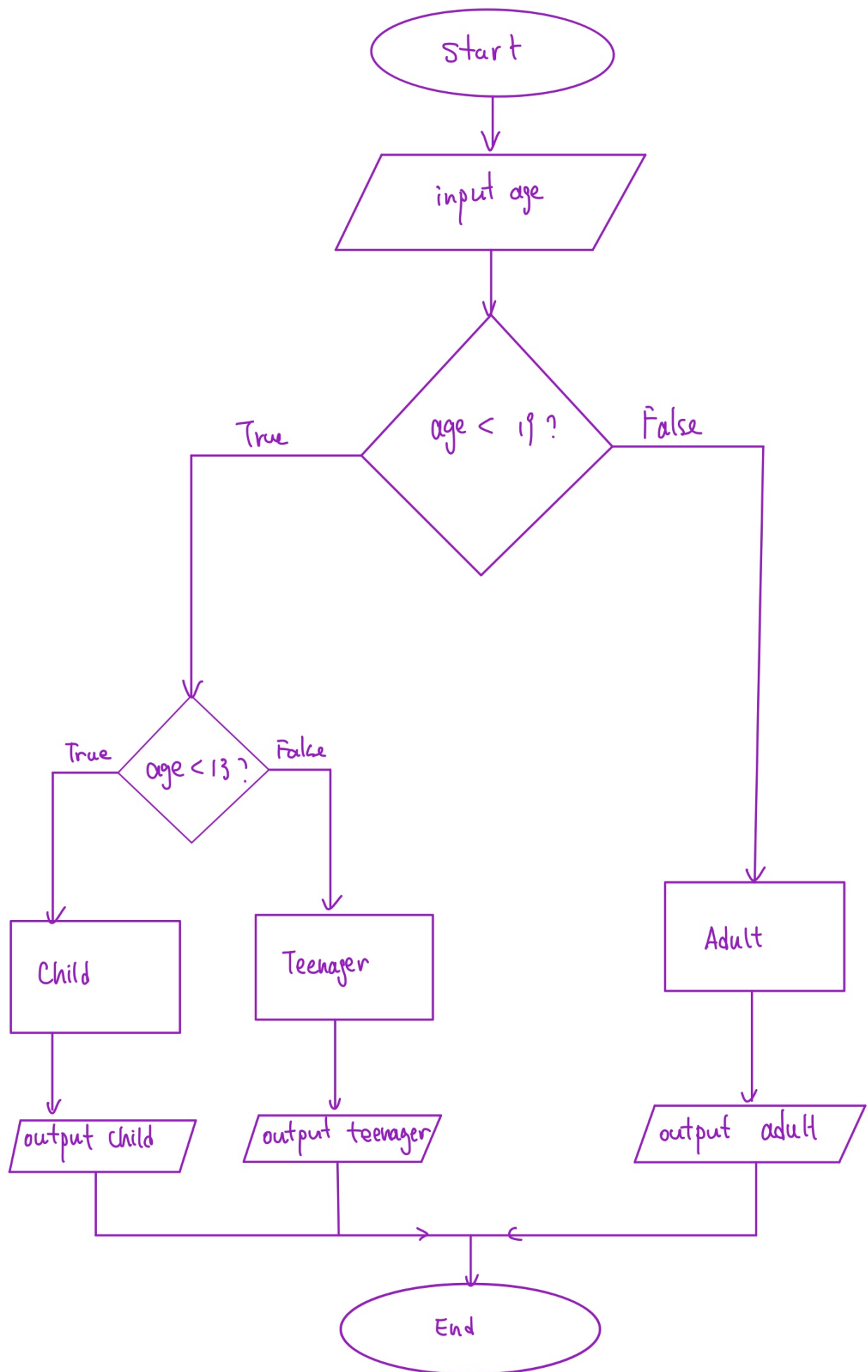


Waiting for grade

3

2 points possible

Give the UML flow chart for deciding whether a given age classifies as an adult, a teenager or a child.



Waiting for grade

4

2 points possible

Describe two common errors discussed in class. Provide example (wrong) code that demonstrates them.

Syntax error, means that the grammar of the language is wrong, or Python does not recognize this grammar. For example, in Python, there is no \$ operator and if we type, say 2\$3, then Python will have no idea what we are doing and return us an error called the Syntax error.

The example can be found below:

```
2$3
```

SyntaxError: invalid syntax

Type error, means that the type of the variables that we are using are wrong. Python does not recognize the type of variables we input while doing operations. For example, you cannot multiply strings using the * multiplication operator.

Example:

```
"hello"*"world"
```

Traceback (most recent call last):

```
File "<pyshell#4>", line 1, in <module>
```

```
"hello"*"world"
```

TypeError: can't multiply sequence by non-int of type 'str'

Waiting for grade

5

2 points possible

Imagine that you are working on code and it is not giving you the output you expected. Describe two things you will do to determine where the errors are happening.

Two things we should do are tracing and debugging. To explain this, when we encounter the problem when the code is not outputting more test cases that shows which part of the program is wrong.

Recall the weighted average example we did in Lesson 12 Module 2.

In that case, Professor Maria wrote up her code with an error after the first attempt, when she calculates the weighted average, she forgot to include all of the categories needed for the problem, (in that case, reading, labs, homework, and exams, all of these four categories needed to be included in the calculation, but she forgot to put a parenthesis, for the purpose of the calculation); instead, professor Maria made a mistake on forgetting to put the parenthesis, this meant that she only calculated the weighted average for "exams" only, but not the other three things (reading, labs, and homework).

Professor Maria traced back to this problem by using four different testing cases, the first one was the test on the "reading" category, the second one was the "labs" category, the third one was "homework" and the last one was the "exams". She found an abnormally large result after testing the "exams" category, which was able to locate and debug the problem, was because she forgot to put the parenthesis, so that Python did the calculation with a

In conclusion, we should always do test cases to test different runs of the problem in order to locate (traceback) where exactly was the problem at our best abilities possible, when our codes are not giving us the output we expected.



Waiting for grade