Calculator

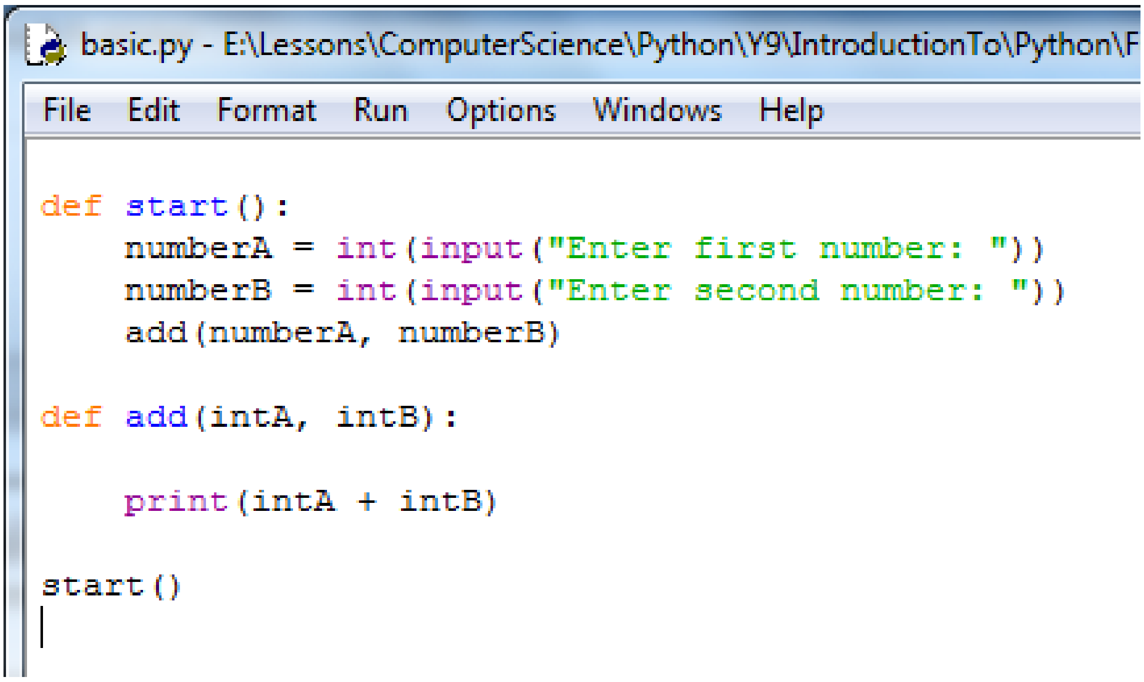
Name:	Class:	Date:

Task 1

Here is the basis for a calculator.

This uses a special programming constructs called a *subroutine.*

The *subroutine* starts with *def* and is very useful when making advance programs. This type of *subroutine* is called a procedure.


basic.py - E:\Lessons\ComputerScience\Python\Y9\IntroductionTo\Python\F
File Edit Format Run Options Windows Help

```
def start():
    numberA = int(input("Enter first number: "))
    numberB = int(input("Enter second number: "))
    add(numberA, numberB)

def add(intA, intB):

    print(intA + intB)

start()
```

Extension 1

Can you create 3 more *subroutines* to multiply, subtract and divide?

In computing we use the following symbols:

- (subtract)

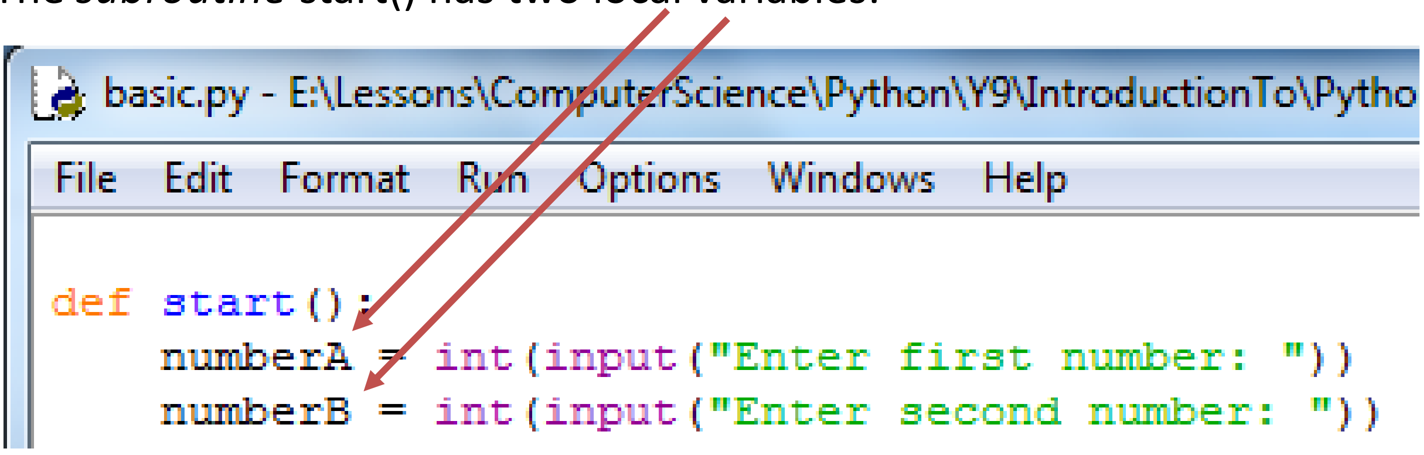
* (Multiply)

/ (Divide)

Make sure you create a skills log as you go.

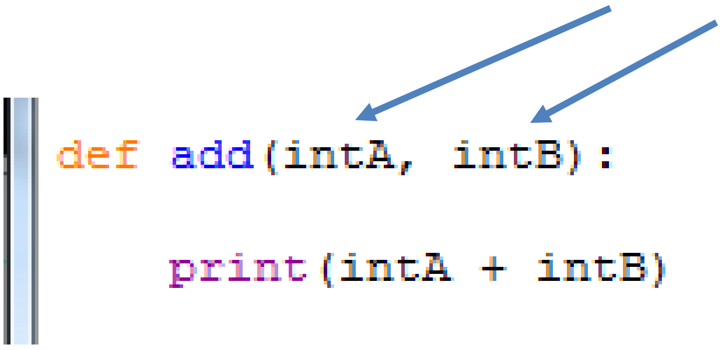
Theory 1

The *subroutine* start() has two local variables:


basic.py - E:\Lessons\ComputerScience\Python\Y9\IntroductionTo\Pytho
File Edit Format Run Options Windows Help

```
def start():
    numberA = int(input("Enter first number: "))
    numberB = int(input("Enter second number: "))
```

The add() *subroutine* has two **parameters**.



```
def add(intA, intB):

    print(intA + intB)
```

When we **call** add() we do it like this:

```
add(numberA, numberB)
```

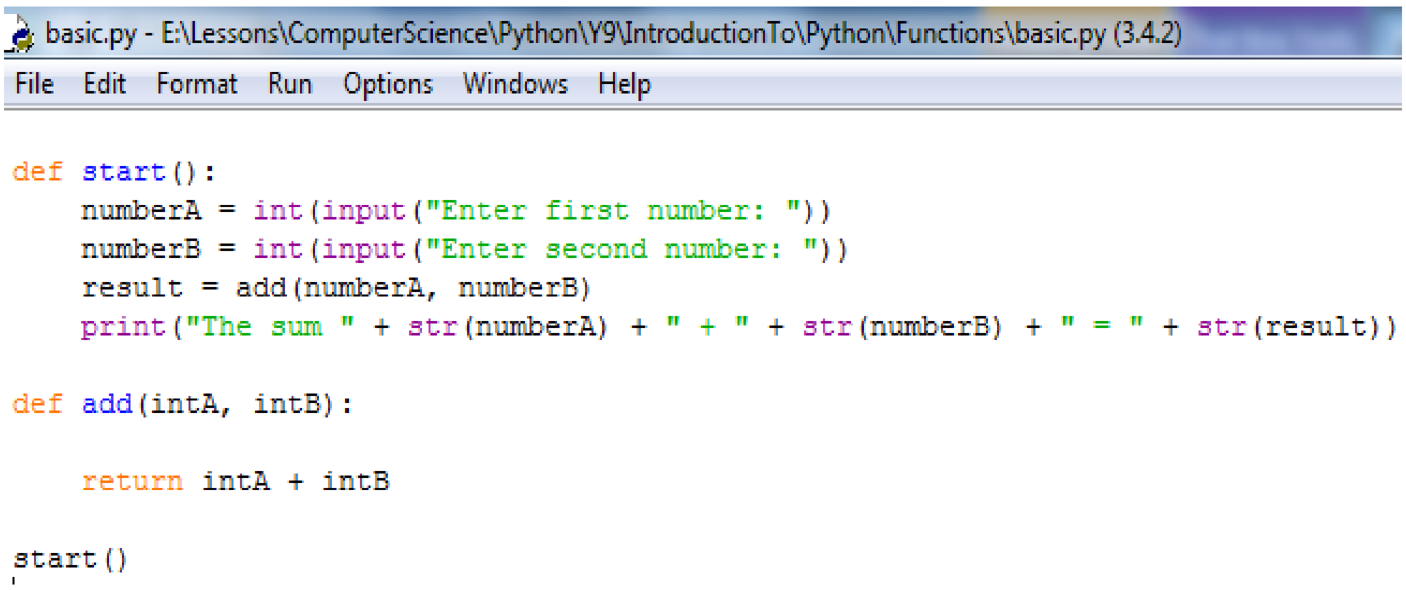
Can you explain the relationship between the variables in start() and the two parameters in add()?

Extension 2

We can improve the code in many ways.

Here I have changed the add() so that instead of printing out it now returns a value.

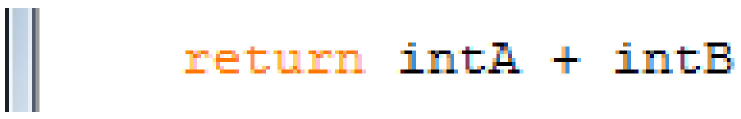
Look carefully at the changes made before writing this code.


basic.py - E:\Lessons\ComputerScience\Python\Y9\IntroductionTo\Python\Functions\basic.py (3.4.2)
File Edit Format Run Options Windows Help

```python
def start():
    numberA = int(input("Enter first number: "))
    numberB = int(input("Enter second number: "))
    result = add(numberA, numberB)
    print("The sum " + str(numberA) + " + " + str(numberB) + " = " + str(result))

def add(intA, intB):

    return intA + intB

start()
```

Theory 2

What does the below line in the add() *subroutine* do?

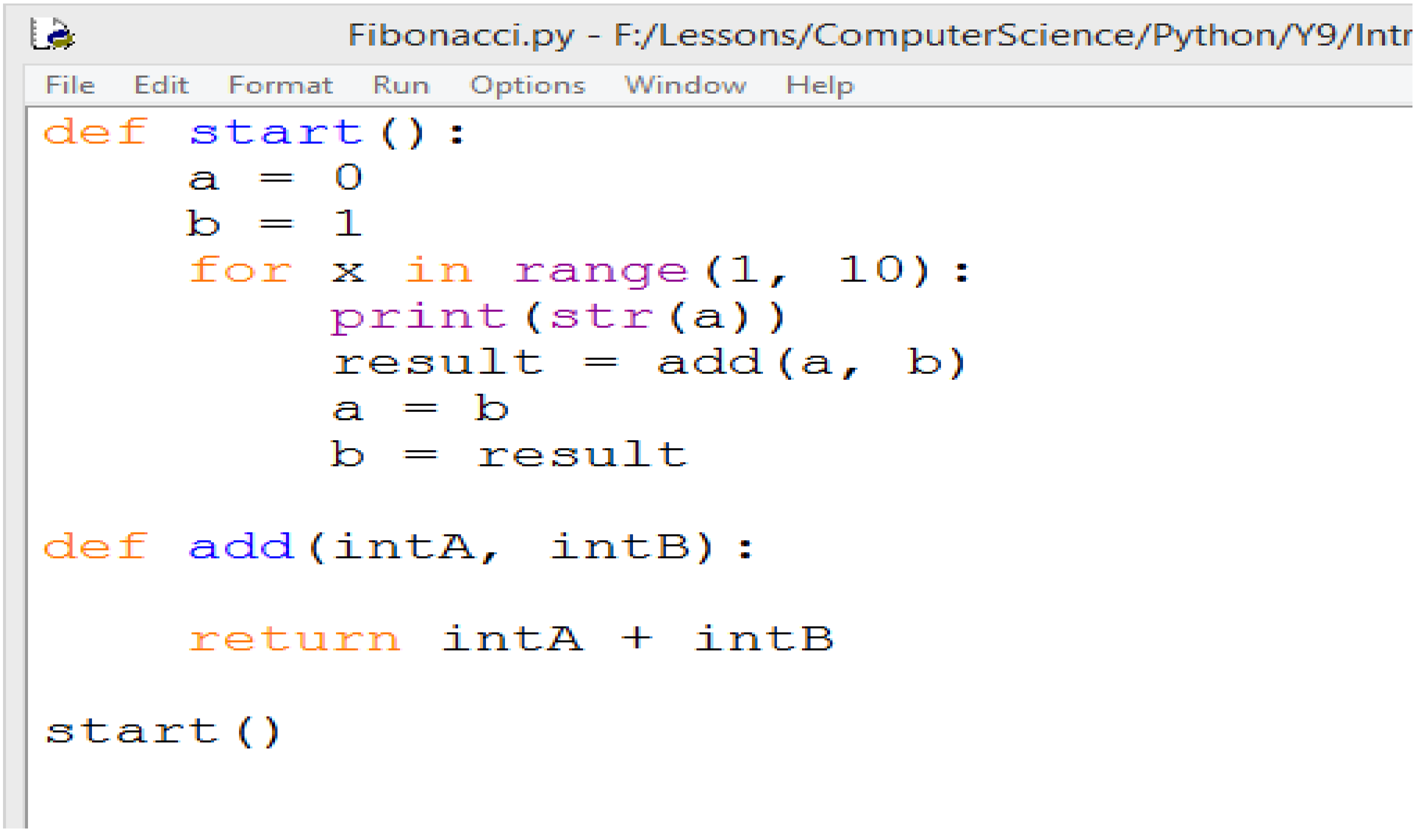


```python
    return intA + intB
```

Task 2

You will now write a program that prints out the Fibonacci sequence.

The sequence goes on forever but we will only go up to the 10th number.


Fibonacci.py - F:/Lessons/ComputerScience/Python/Y9/Intr
File Edit Format Run Options Window Help

```
def start():
    a = 0
    b = 1
    for x in range(1, 10):
        print(str(a))
        result = add(a, b)
        a = b
        b = result

def add(intA, intB):

    return intA + intB

start()
```

Extension 1

The For loop stops once it *iterates* 10 times. Can you make it so a user can **input** their own number of iterations?

Begin by seeing what happens when you make this value

higher or lower.

```
for x in range(1, 10):
```

You may want a variable with a name such as: **maxNum**