Exercises

What would be the output from each of the following lines of code?

```
1  print(1+2+3+4+5)
2  print (3**2 + 4**2)
3  print(3*(5+2))
4  #print(100)
5  print(1/2 + 1/2)
6  print (1//2 + 1//2)
7  print (3985780149 % 2)
```

What would be the code you would write for each of the following?

8. To find the number of items in 8 dozen
9. To find and print the number of weeks in 180 days
10. To print out “I love Python!”
11. A comment to indicate that it is your first program

Exercises

For each of the following short programs, what would be the output of the segment of code?

1
```
a = 10
b = 15
a += b
print(a)
```

2
```
a = 10
b = 15
a = b
b = 1
print(a)
```

3
```
a = 10
b = 15
a = a*a+b
print(a)
```

4
```
a = 10
b = 15
a *= a+b
print(a)
```

5
```
a = 10
b = float(a)
print(b)
```

6
```
a = "10"
b = int(a)
print(b)
```

7
```
a = "Welcome"
b = "Home"
print(a,b)
```

8
```
a="Welcome"
b="Home"
print(a+b)
```

9
```
a = "10"
b = "15"
c = a+b
d = int(c)
print(d)
```

What code would you write for each of the following?

10 Set the price of bread to be 2.00.

11 Given a price for a loaf of bread, "bread_price," and a price for a block of cheese, "cheese_price," calculate the cost to buy 2 loaves of bread and 3 blocks of cheese.

12 Get a user's age.

13 Write a program to form the name of a knight by asking the user for the knight's name and a personality characteristic. The final name should be printed as "Sir <name> the <characteristic>." For example, if the user enters "Robin" and "Brave," you would print "Sir Robin the Brave."

›

Exercises

Assume that you have the following lines of code.

```
a = 1
b = 2
c = 2
d = "One"
e = "Two"
f = "Three"
g = "one"
```

Would these Boolean expressions be true or false?

```
1  a > b
2  a == b
3  a != b
4  b == c
5  d < e
6  e < f
7  d < g
8  g < e
```

9 `not (a == b)`

10 `b < c or b > c`

11 `(a+1) == b and not b < c`

12 `((a <= b) and (b <= c)) or ((a >= b) and (b >= c))`

What would be the output for each of the following segments of code?

13
```
total_cost = 100.00
days = 3
cost_per_day = total_cost / days
if cost_per_day > 40:
    print("Too expensive")
elif cost_per_day > 30:
    print("Reasonable cost")
elif cost_per_day > 20:
    print("Excellent cost")
else:
    print("Incredible bargain")
```

14
```
age = 67
income = 10000
if (age > 70):
    if (income < 15000):
        print("Eligible for benefits")
    else:
        if (income < 20000):
            print("Eligible for reduced benefits")
        else:
            print("Not eligible for benefits")
else:
    if (income < 20000):
        print("Eligible for reduced benefits")
    else:
        print("Not eligible for benefits")
```

Write code for each of the following.

15 Rewrite the code in exercise 14 in a simpler way by using a more complex Boolean expression and an elif statement.

16 Compare a variable “user_guess” to a variable “hidden_answer,” and tell the user whether the guess is too low, too high, or exactly right.

17 Generally, every fourth year is a leap year, but there are exceptions. If the year is divisible by 100, then it is not a leap year, unless the year is also divisible by 400, in which case it is still a leap year. So, 2000 (divisible by 400) is a leap year, 2100 (divisible by 100 but not 400) is not, 2004 (divisible by 4 but not 100) is a leap year, and 2001 (not divisible by 4) is not. Write code that examines a variable and year and prints out “Leap year” or “Not a leap year” for that value. Try writing the code in the following three different ways.

a) As a series of nested if statements
b) As a set of if-elif-else statements
c) As a single if statement with a complex Boolean expression

›

›

Exercises

Try making additional iterations to improve the code developed in the lecture. The following are a few possible improvements you might want to make.

1. Assume that the user has already saved up some amount of money. Ask the user for an amount already saved. (Note that the balance will be the cost minus the amount already saved.)

2. Ask the user for the period (week, month, etc.) for how often they will regularly save money. Use this to make the input and output for the user more meaningful.

Exercises

What would be the output of the following code?

1
```
i = 10
while i > 1:
    print (i)
    i /= 2
```

2
```
i = 0
value = 0
while value < 20:
    value += i
    i += 1
    print(value)
```

3
```
for i in range(4):
    print (i)
```

4
```
for i in range(3,5):
    print (i)
```

5
```
for i in range (1,10,3):
    print (i)
```

6
```
for i in range (1, 10, -3):
    print (i)
```

7
```
for i in range (10, 1, -3):
    print (i)
```

Write code to do each of the following.

8 Get a number from the user, and then count from 1 to that number. Try writing it using both a while loop and a for loop.

9 Convert the following while loop into a for loop.

```
i = 2
while(i<7):
    print(i)
    i = i + 3
```

10 Write a short program that defines a number from 1 to 10, and then keeps asking the user to guess that number until the correct number is guessed.

Exercises

```
1   mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
    print (mylist[1])

2   mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
    print (mylist[2:5])

3   mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
    print (mylist[:3])

4   mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
    print (mylist[8:])

5   mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
    print (mylist[:])

6   mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
    print (mylist[-1])

7   mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
    print (mylist[-3:])

8   mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
    print (mylist)
```

9
```
mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
for i in mylist:
    print (i)
```

10
```
mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
mylist[3] = 100
print (mylist)
```

11
```
mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
for i in mylist:
    i = 0
print (mylist)
```

12
```
mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
mylist.append(100)
print(mylist)
```

13
```
mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
mylist[1:5] = []
print (mylist)
```

14
```
mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
mylist[2:8] = [100, 200]
print (mylist)
```

15
```
mylist = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
mylist [2:2] = [100]
print (mylist)
```

Write code to do the following:

16 Given a list of integers named “ages,” form a new list named “minor_ages” consisting of all those ages from the “ages” list that are less than 18.

17 Create a list containing two lists: one with names of 3 people and one with ages of 3 people (you can choose the names/ages).

Exercise

Modify the program developed in this lecture to also keep track of the chance that there will be rain on the particular day. Below are a few hints if you need them.

- ◊ You already are reading in the data you need to.
- ◊ You will need to add some additional lines to the analysis and presentation parts of the code.
- ◊ Keep track of how many days had rain as you go through the list "gooddata."
- ◊ Compute a percentage of days with rain and report that.

›

Exercises

What would be the output of the following code?

1
```
def something1(a, b):
    for i in range(a):
        print(b,end='')
something1(5, 'X')
```

2
```
def something2(a, b):
    for i in range(a):
        b = b*b
    return b
print(something2(3,2))
```

3
```
def something3(a):
    return a-1, a+1
a, b = something3(5)
print(a, b)
```

4
```
def something4(a):
    sum = 0
    for b in a:
        if b < 0:
            sum -= b
        else:
            sum += b
    return sum
print(something4([2, -4, 3, -1, 7, -4]))
```

5
```
def something5(a):
    sum1 = 0
    sum2 = 0
    for i in range(len(a)):
        if i%2 == 0:
            sum1 += a[i]
        else:
            sum2 += a[i]
    return sum1, sum2
x, y = something5([1, 2, 3, 4, 5, 6])
print(x,y)
```

Write code for the following.

6 A function that takes in a number and a string and prints the string that many times.

7 A function that takes in two lists of the same length and returns a new list of that length, containing the smaller of the elements at that index value from the two lists.

8 A function that takes in three numbers and returns the one in the middle.

Simplify the following code using a function.

9
```
salary1 = float(input("Enter previous salary"))
benefits1 = float(input("Enter previous benefits"))
bonus1 = float(input("Enter previous bonus"))
salary2 = float(input("Enter new salary"))
benefits2 = float(input("Enter new benefits"))
bonus2 = float(input("Enter new bonus"))
if salary2 > salary1:
    salaryincrease = salary2 - salary1
else:
    salaryincrease = 0
if benefits2 > benefits1:
    benefitsincrease = benefits2 - benefits1
else:
    benefitsincrease = 0
if bonus2 > bonus1:
    bonusincrease = bonus2 - bonus1
else:
    bonusincrease = 0
```

›

›

Exercises

What would be the output of the following code?

```
1   def something1(a):
        a = 0
    b=3
    something1(b)
    print(b)

2   def something2(a):
        a[0] = 0
    b=[1,2,3]
    something2(b)
    print(b)
```

3
```
def something3(a, b=2, c=3, d=4):
    return a + b + c + d
val = something3(3, 10, d=5)
print(val)
```

4
```
def something4():
    a = 3
a = 2
something4()
print(a)
```

5
```
def something5():
    global a
    a = 3
a = 2
something5()
print(a)
```

6
```
def something6():
    a[0] = 0
a = [1, 2, 3]
something6()
print(a)
```

7
```
def something7(a, b):
    print (a, b)
a = 1
b = 2
something7(b, a)
```

Write code for the following.

8 A function that increases all the elements of a list by 1.

9 A function that multiplies anywhere from 1 to 4 parameters together, returning the product of those numbers.

›

Exercises

1 Imagine that you have written a piece of code that is supposed to return a ticket price given an age. Those under age 3 are free, other children from 3 to 12 are $5, and all others are considered adults and cost $10. When you test your code, what are the ages that would be good to use in your tests?

2 Assume that you have written the following code to find the middle element from a 3-element list.

```
def findmiddle(a):
    if ((a[0] >= a[1]) and (a[1] >= a[2])) or ((a[0] <= a[1])
      and (a[1] <= a[2])):
        return a[1]
    elif ((a[0] >= a[2]) and (a[2] >= a[1])) or ((a[0] <=
      a[2]) and (a[2] <= a[1])):
        return a[2]
    else:
        return a[0]
```

Notice that if a list is passed in that is not of length at least 3, the code will give an error.

a) Modify the function so that it will raise an exception if the list is not valid.

b) Then, show how you could call the function, printing a message if there was an exception.

Exercises

1 Assume that the “Stack” class is defined as in the lecture. What would be the output of the following code?

```
namestack = Stack()
namestack.push("John")
namestack.push("James")
namestack.push("Joseph")
person = namestack.pop()
print(person)
person = namestack.pop()
print(person)
person = namestack.pop()
print(person)
```

2 Assume that the “Queue” class is defined as in the lecture. What would be the output of the following code?

```
namequeue = Queue()
namequeue.enqueue("John")
namequeue.enqueue("James")
namequeue.enqueue("Joseph")
person = namequeue.dequeue()
print(person)
person = namequeue.dequeue()
print(person)
person = namequeue.dequeue()
print(person)
```

3 What would be the output of the following code?

```
cast = {"Cardinal Ximenez" : "Michael Palin", "Cardinal
   Biggles" : "Terry Jones", "Cardinal Fang" : "Terry
   Gilliam"}
cast["customer"] = "John Cleese"
cast["shopkeeper"] = "Michael Palin"
print(cast["shopkeeper"])
print(cast["Cardinal Ximenez"])
print(cast["Cardinal Fang"])
```

4 What would be the output of the following code?

```
primes = {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37}
teens = set([13, 14, 15, 16, 17, 18, 19])
print(primes - teens)
print(primes & teens)
print(primes | teens)
print(primes ^ teens)
```