Chapter 6 Functions

1. At least three benefits: (1) Reuse code; (2) Reduce complexity; (3) Easy to maintain.

2. See the Sections 6.2 and 6.3 on how to declare and invoke functions.

3. `return num1 if (num1 > num2) else num2`

4. True: a call to a function with a void return type is always a statement itself.
   False: a call to a value-returning function is always a component of an expression.

5. A syntax error occurs if a return statement is not used to return a value in a value-returning function. You can have a return statement in a void function, which simply exits the function. So, this function is OK, although the return statement is not needed at all.

6. See Section 6.2.

7. Computing a sales commission given the sales amount and the commission rate
   def getCommission(salesAmount, commissionRate):
   returns a value

   Printing a calendar for a month
   def printCalendar(month, year):

   Computing a square root
   def sqrt(value):
   returns a value

   Testing whether a number is even and return true if it is
   def isEven(value):
   returns a value

   Printing a message for a specified number of times
   def printMessage(message, times):

   ```
   Computing the monthly payment, given the loan amount,
   number of years, and annual interest rate.

   def monthlyPayment(loanAmount, numberOfYears,
   annualInterestRate):
   ```

   returns a value

Finding the corresponding uppercase letter given a lowercase letter.
def getUpperCase(char letter):
returns a value

8. Line 2, 5-10: not indented correctly.

9. None

10. The min function should return a value.

11. Using positional arguments requires the arguments be passed in the same order as their respective parameters in the function header. You can also call a function using *keyword arguments*, passing each argument in the form *name=value*.

12.

```
f(1, p2 = 3, p3 = 4, p4 = 4)  # Correct
f(1, p2 = 3, 4, p4 = 4) # Not Correct
f(p1 = 1, p2 = 3, 4, p4 = 4) # Not Correct
f(p1 = 1, p2 = 3, p3 = 4, p4 = 4) # Correct
f(p4 = 1, p2 = 3, p3 = 4, p1 = 4) # Correct
```

13. "Pass by value" is to pass a copy of the reference value to the function.

14. Yes.

15.

(A) The output of the program is 0, because the variable max is not changed by invoking the function max.

(B)

2
2 4
2 4 8
2 4 8 16
2 4 8 16 32

(C)

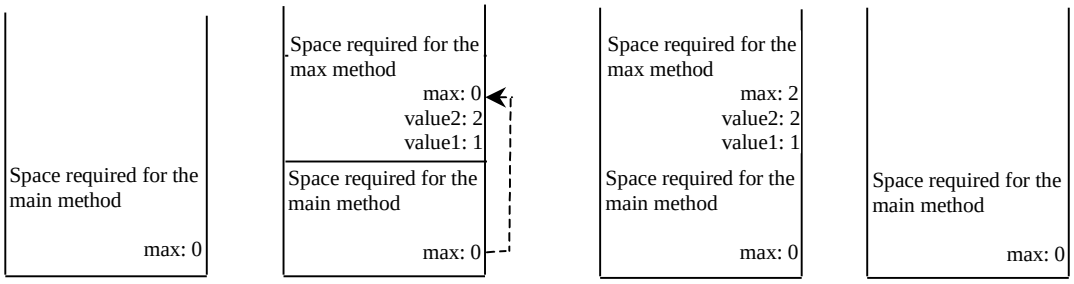
Before the call, variable times is 3
n = 3
Welcome to CS!
n = 2
Welcome to CS!
n = 1
Welcome to CS!
After the call, variable times is 3

(D)
The code prints

```
i is 1
 1
i is 2
 2 1
i is 3
```

and then it is an infinite loop.

16.



Just before max is invoked.

Just entering max.

Just before max is returned

Right after max is returned

17.

(A)

```
2
3.4
2
4
```

(B)

```
3
6
5
```

18.

x and y are not defined outside the function.

19.

Yes. The printout is

```
y is 1
```

20.

```
1 2
5 2
1 24
4 5
```

21.

The parameter with non-default Argument must be defined before the parameter with default arguments.

22.

The later function definition will replace the previous function definition.

23.

Yes. The printout is

```
14 4 45 1.8
```

24.

```
random.randint(34, 55)
```

25.

```
chr(random.randint(ord('B'), ord('M')))
```

26.

```
5.5 + random.random() * 50
```

27.

```
chr(random.randint(ord('a'), ord('z')))
```