|  |  |
| --- | --- |
|  | Chapter 2 Practice Questions – Flow Control |
| **Q1** | **What are the two values of the Boolean data type? How do you write them?** |
| A | Boolean data type has only two values: True and False |
| **Q2** | **What are the three Boolean operators?** |
| A | The three Boolean operators (and, or, and not) are used to compare Boolean values |
| **Q3** | **Write out the truth tables of each Boolean operator (that is, every possible combination of Boolean values for the operator and what they evaluate to)** |
| A | The and operators’ truth table:  True and True = True  True and False = False  False and True = False  False and False = False  The or operator’s truth table:  True or True = True  True or False = True  False or True = True  False or False = False  The not operators’ truth table:  Not True = False  Not False = True |
| **Q4** | **What do the following expressions evaluate to?**   1. **(5 > 4) and (3 == 5)** 2. **Not (5 > 4)** 3. **(5 > 4) or (3 == 5)** 4. **Not ((5 > 4) or (3 ==5))** 5. **(True and True) and (True == False)** 6. **(not False) or (not True)** |
| A | 1. False 2. False (The not operator simply evaluates to the opposite Boolean value) 3. True 4. False 5. False 6. True |
| **Q5** | **What are the six comparison operators?** |
| A | == | Equal to  != | Not equal to  < | Less than  > | Greater than  <= | Less than or equal to  >= | Greater than or equal to |
| **Q6** | **What is the difference between the equal to operator and the assignment operator?** |
| A | The equal to operator asks whether two values are the same as each other and evaluate to a Boolean value of True or False  The assignment operator puts the value on the right into the variable on the left |
| **Q7** | **Explain what a condition is and where you would use one** |
| A | Conditions are expressions which always evaluate down to a Boolean value (True or False) |
| **Q8** | **Identify the three blocks in this code:**  **spam = 0**  **If spam == 10:**  **print(‘eggs’)**  **if spam > 5:**  **print(‘bacon’)**  **else:**  **print(‘ham’)**  **print(‘spam’)**  **print(‘spam’)** |
| A | 3; separated by the colon  The colon is there to declare the start of an indented block |
| **Q9** | **Write code that prints Hello if 1 is stored in spam, prints Howdy if 2 is stored in spam, and prints Greetings! If anything else is stored in spam.** |
| A | if spam == 1:  print(‘Hello’)  elif spam == 2:  print(‘Howdy’)  else:  print(‘Greetings!’) |
| **Q10** | **What can you press if your program is stuck in an infinite loop?** |
| A | If you ever run a program that has a bug causing it to get stuck in an infinite loop, press CTRL-C |
| **Q11** | **What is the difference between break and continue?** |
| A | The difference is that the break statement causes a while loop to terminate based on a set condition(s) being met whereas the continue statement also terminates a while loop until a set condition(s) is met but if the condition is not met it goes back to the beginning of the while loop to retest the condition(s) unlike the break statement which does not return the program to the beginning of the while loop if the condition(s) is not met. |
| **Q12** | **What is the difference between range(10), range(0, 10), and range(0, 10, 1) in a for loop?** |
| A | A block of code can be run a specified number of times using the for loop statement e.g.  For i in range(5):  print(‘Hello’)   1. range(10) will run a block of code 10 times with each sequence being 0-9 2. range(0, 10) will run a block of code a block of code 10 times (10 not inclusive). These two arguments allow for the range iterations to start from any number other than 0 if so wish 3. range(0, 10, 1) will run a block of code 10 times. The third argument is by how many numbers the loop should skip e.g. range(0, 10, 2) will skip each sequence by 1 and each sequence will thus be: 0, 2, 4, 6 and 8 |
| **Q13** | **Write a short program that prints the numbers 1 to 10 using a for loop. Then write an equivalent program that prints the numbers 1 to 10 using a while loop.** |
| A | For loop:  for I in range(1, 11):  print(i)  While loop:  I = 1  while I <= 10:  print(i)  I = I + 1 |
| **Q14** | **If you had a function named bacon() inside a module named spam, how would you call it after importing spam?** |
| A | import spam  spam.bacon() |