

Chapter 5 is “Picture Techniques with Selection and Combination.” The “Selection” part is accomplished via `if` statements. An `if` statement allows code to be executed (i.e., “selected to execute”) or not based on the result of a comparison. The syntax of an `if` statement is:

```
if <condition>:
    statement1
    statement2
    statement3
```

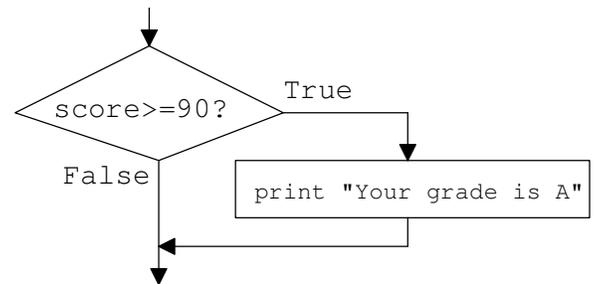
If the condition evaluates to `True`, then the statements of the indented body are executed. If the condition is `False`, then the body is skipped.

Typically, the condition involves comparing “stuff” using relational operators (`<`, `>`, `==`, `<=`, `>=`, `!=`). For example, we might want to print “Your grade is A.” if the variable `score` is greater-than or equal to 90.

```
if score >= 90:
    print "Your grade is A."
```

Complex conditions might involve several comparisons combined using Boolean operators: `not`, `or`, `and`. For example, we might want to print “Your grade is B.” if the variable `score` is less than 90, but greater than or equal to 80.

```
if score < 90 and score >= 80:
    print "Your grade is B."
```



1. Using only `if` statements, complete the Python code to output the appropriate string according to variable `score`'s value using the grading scale:

score	String
<code>score &gt;= 90</code>	“Your grade is A.”
<code>80 &lt;= score &lt; 90</code>	“Your grade is B.”
<code>70 &lt;= score &lt; 80</code>	“Your grade is C.”
<code>60 &lt;= score &lt; 70</code>	“Your grade is D.”
<code>score &lt; 60</code>	“Your grade is F.”

```
if score >= 90:
    print "Your grade is A."
```

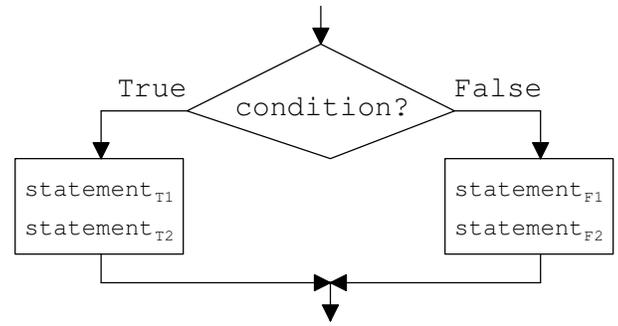
```
if score < 90 and score >= 80:
    print "Your grade is B."
```

2. An `if-else` statement allows a block of code to be executed if the result of a comparison is `True`; otherwise the “else” block of code will be executed. The syntax of an `if-else` statement is:

```
if <condition>:
    # only executed if condition is True
    statementT1
    statementT2
else:
    # only executed if condition is False
    statementF1
    statementF2
```

Using only nested `if-else` statements, complete Python code to output the appropriate string according to variable `score`'s value.

```
if score >= 90:
    print "Your grade is A."
else:
    if score >= 80:
        print "Your grade is B."
    else:
```



3. The `if-elif-else` statement is useful for implementing nested if statement like above. Complete Python code to output the appropriate string according to variable `score`'s value.

```
if score >= 90:
    print "Your grade is A."
elif score >= 80:
    print "Your grade is B."
elif
```

4. The precedence for mathematical operators, Boolean operators, and comparisons are given in the table. For each of the expressions, determine if the expression is legal. If it is, determine the order of operations:

Operator(s)		
highest ↑  lowest	+, - (unary)	a) $x + 5 < y * 2$ and not $y \geq 6$
	** (exponent)	
	*, /, %	
	+, - (add, sub)	
	<, >, ==, <=, >=, !=	b) not $x < 8$ or $y / 3 \geq 10$
	not	
	and	
	or	

5. In Chapter 5 the `if` statements we write will typically be checks to see if two colors are close enough. JES comes with a `distance(color1, color2)` function that takes two `Color` objects and returns a single number representing the distance between the colors. The red, green, and blue values of the colors are taken as a point in  $(x, y, z)$  space, and the Cartesian distance is computed. For example, the textbook red-eye reduction:



```

""" Red-eye reduction in a rectangle region of picture """
def main():
    print "Select the Media Folder"
    setMediaFolder()
    print "Select the picture (jenny-red.jpg) file"
    fileName = pickAFile()
    pict = makePicture(fileName)

    print "Please wait while picture is processed."
    removeRedEye(pict, 109, 91, 202, 107, black)

    repaint(pict)

def removeRedEye(pict, startX, startY, endX, endY, replacementColor = black):
    """ Remove red in a range """

    red = makeColor(255, 0, 0)
    for x in range(startX, endX+1):

        for y in range(startY, endY+1):
            currentPixel = getPixel(pict, x, y)
            if distance(red, getColor(currentPixel)) < 165:
                setColor(currentPixel, replacementColor)

main() # starts the program

```