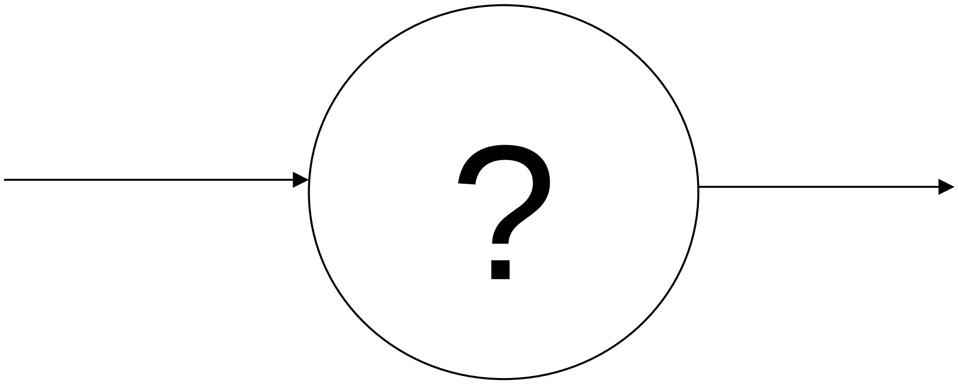
Process Specification

Lowest-level DFD Process



How does this process work?
What tools can be used to show the internal, step-by-step workings of this process?

Process Specification tools:

- ➔ Procedural: How To
  - Structured English
  - Flowchart

- ➔ Modal: When to
  - Decision table
  - Decision tree

YOURDON PRESS

ORDER FORM

Please check this box if you are a new customer:

- □ <tentative-customer-ID>

Name: <customer-name>

Company Name: <company-name>

Address: <customer-address>

Book Code	Quantity	Unit Price	Discount	Extended Price
<book-code>	<quantity-ordered>	<unit-price>	<discount>	<extended-price>
.....*ABC*......	......*3*......	......*$30*......	......*10%*......	......
.....*XYZ*......	......*5*......	......*$20*......	*7%*......	......

Subtotal: <subtotal>

8% ← Sales Tax Rate: <sales-tax-rate>

$6 ← Shipping Charges: <shipping-charges>

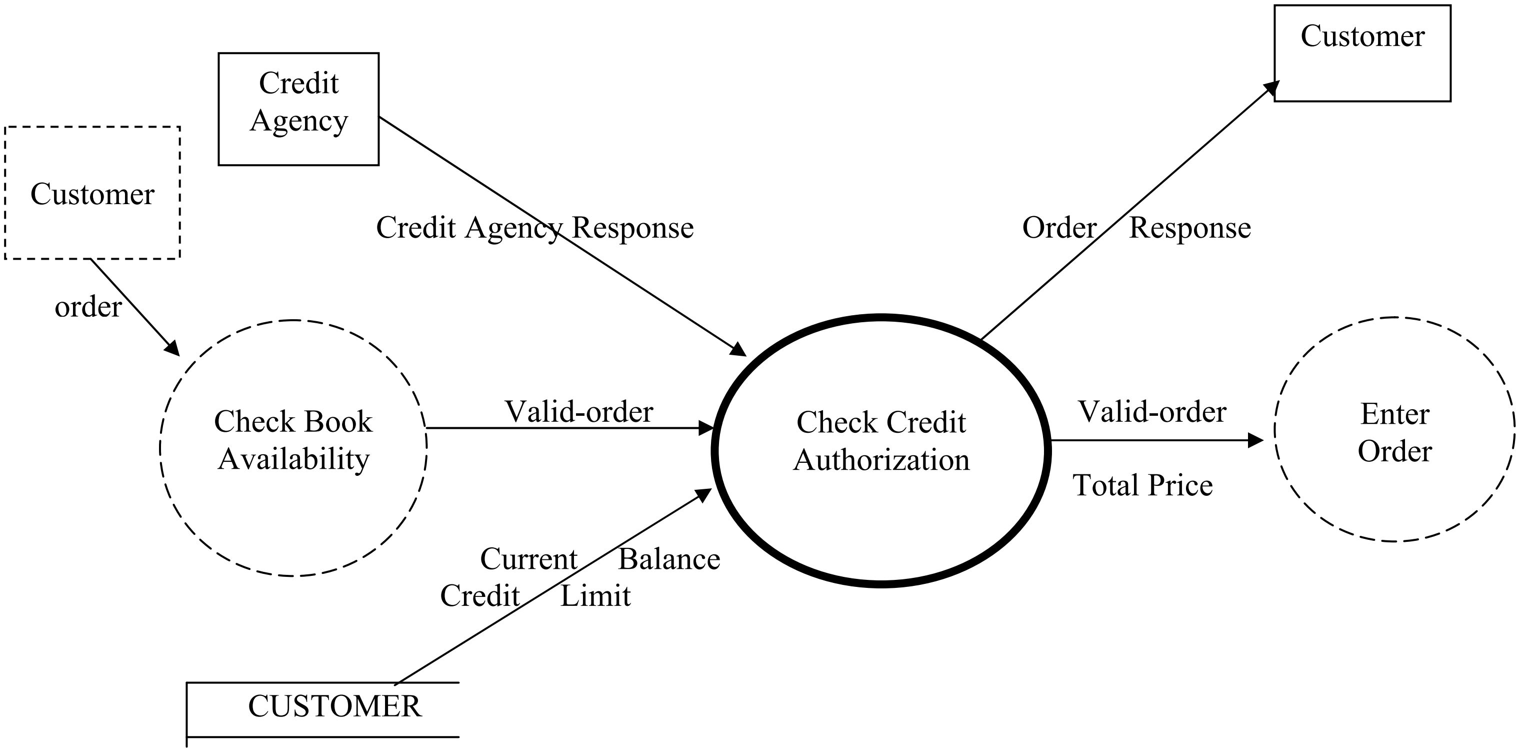
Total: < total-price>

How would you like to pay for this order? <payment-type>

- □ Cash → *$190*
- □ Check → *$190*
- □ Credit card → *credit limit on credit card = $150*
- □ Bill me → *current balance = $500, credit limit = $600*

If cash or check, total amount of payment enclosed:<order-payment> (*cash/check = $190*)

Consider the following process performed by the order entry department of a publisher. The task being performed here is authorizing a customer's credit for an order after that order has been validated in terms of the availability of the book requested. The DFD appears below.



Unstructured Specification for Process “Check Credit Authorization”

For each line of the valid order, the extended price is calculated by multiplying the quantity ordered by unit price and by discount. These extended prices are added to produce the subtotal. This subtotal is multiplied by the sales tax rate to calculate the tax. The subtotal is added to the tax and the shipping charges to come up with the grand total price of the order. The action taken regarding the customer's order from this point on depends on the type of payment requested. In case the payment type selected by the customer is "cash" or "check", if the total-price (computed above) is greater than payment enclosed, then the customer is sent the message "Purchase price exceeds amount paid", and no further processing of the order will take place. In case the payment type selected by the customer is "credit card", a response is requested from the credit agency. If it is "No", then the customer is sent the message “Credit request denied", and no further processing of the order will take place. In case the payment type selected by the customer is "Bill me", the customer’s balance is read and added to the total price of the order. If this sum exceeds the customer’s credit-limit, then the customer is sent the message "Order exceeds your credit limit", and no further processing of the order will take place. If the processing of the order has not been suspended by this point, the order details as well as the total price are passed on to the next process, "enter order".

Structured Specification for Process “Check Credit Authorization”

```
BEGIN

* Calculating the order subtotal
subtotal = 0
DOWHILE there are more order-items in valid-order
    GET quantity-ordered, unit-price, discount FROM "Check Book Availability"
    extended-price = quantity-ordered * unit-price * (1 – discount)
    subtotal = subtotal + extended-price
ENDDO

tax = subtotal * sales-tax-rate
total-price = subtotal + tax + shipping charges
```

Or, combine the two steps into one:
total-price = subtotal * (1 + sales-tax-rate) + shipping-charges

```
DOCASE
    CASE payment-type = "cash" or "check"
        IF total-price > order-payment THEN
            order-response = "Purchase price exceeds amount paid"
            PRODUCE order-response TO customer
            EXIT
        ENDIF
    CASE payment-type = "credit card"
        ACCEPT credit-agency-response FROM credit agency
        IF credit-agency-response = "No" THEN
            order-response = "Credit request denied"
            PRODUCE order-response TO customer
            EXIT
        ENDIF
    CASE payment-type = "Bill me"
        FIND RECORD IN CUSTOMER WITH MATCHING customer-id IN valid-order
        READ current-balance, credit-limit FROM CUSTOMER
        IF current-balance + total-price > credit-limit THEN
            order-response = "Order exceeds your credit limit"
            PRODUCE order-response TO customer
            EXIT
        ENDIF
ENDCASE

SEND valid-order and total-price TO “Enter Order”

END
```

A Process Specification Language

To declare the beginning of a program: BEGIN ○

INPUT

From an external entity: ACCEPT <name of data> FROM <name of external entity>

- ACCEPT customer-complaint FROM customer

From another process: GET <name of data> FROM <name of process>

- GET valid-customer-complaint FROM “Validate Customer Complaint”

From a data store:
STEP 1: Search for a matching record:

FIND RECORD IN <file name> WITH MATCHING <field name> IN <source>

- FIND RECORD IN customer WITH MATCHING customerid IN valid-customer-complaint

Or, more formally:

FIND RECORD IN <file name> WHERE <file name>.<field name> = <source>.<field name>

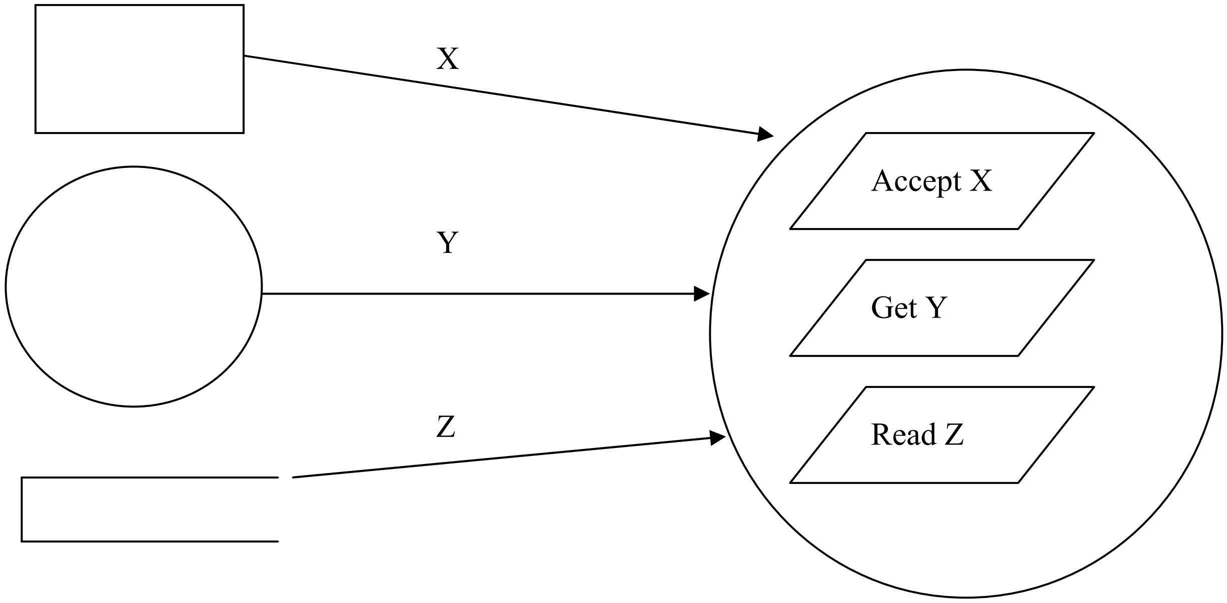
- FIND RECORD IN customer WHERE customer.customerid = valid-customer-complaint.customerid

STEP 2: Get the value of a field in a specific record: READ <field name> FROM <file name>

- READ balance FROM customer

Or, combine the two steps into one:

- READ balance FROM customer WITH MATCHING customerid IN valid-customer-complaint

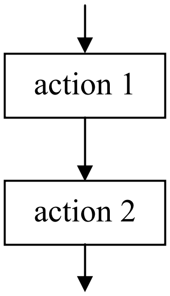


PROCESSING

Control Structures: When to do what

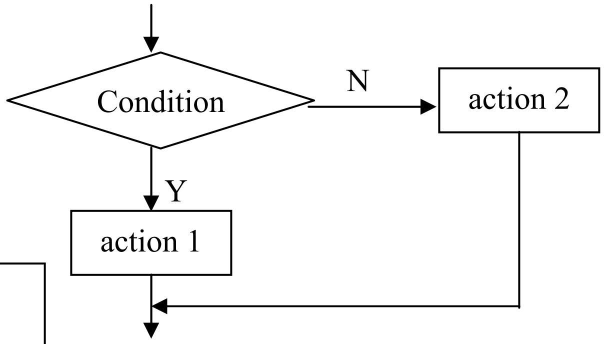
Sequence:

```
        <action 1>
        <action 2>
```



Selection (condition):

```
        IF <condition> THEN
                <action 1>
        ELSE
                <action 2>
        ENDIF
```



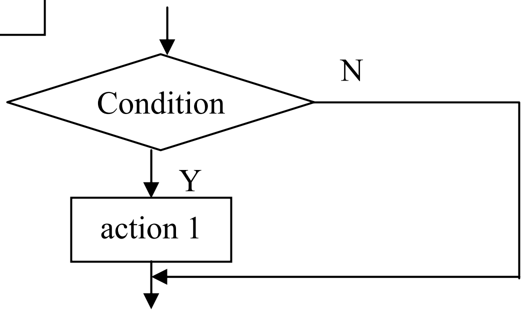
Not Acceptable

```
IF <condition> THEN
        <action 1>
ENDIF

IF <NOT condition> THEN
        <action 2>
ENDIF
```

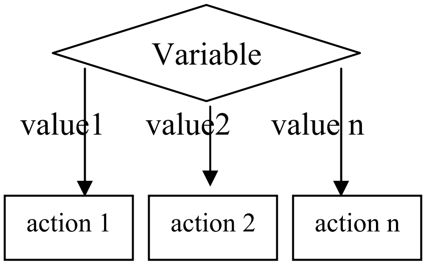
Special case: no else

```
        IF <condition> THEN
                <action>
        ENDIF
```



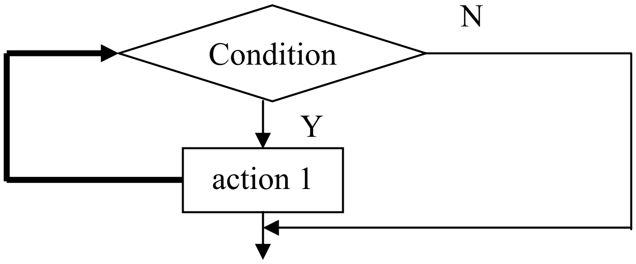
Special case: various values of a variable

```
        DOCASE
                CASE <variable = value 1>
                        <action 1>
                CASE <variable = value 2>
                        <action 2>
                ................................
                CASE <variable = value n>
                        <action n>
        ENDCASE
```



Iteration (repetition/looping):

```
DOWHILE <condition>
        <action 1>
ENDDO
```



Other variations: DOUNTIL, DOFOR

Note:

- Indent properly as shown above
- Actions, as well as conditions, can be composite: (A and B), (A or B), (not A)
- Nesting is possible (selection within selection, selection within iteration, etc.)
- A special case is EXIT: suspend all further processing; end the program

Not Acceptable

```
IF <condition> THEN
        <action>
        EXIT
ENDIF
```

≠

```
IF <condition> THEN
        <action>
ENDIF
```

Think About!

Sequence = Sequence
Selection = Conditional Sequence
Iteration = Repetitive Sequence

Operations: Producing Transformations

Logical Operations

- Comparison (= > < ≥ ≤ ≠)
- Logical operators (NOT, AND, OR)

Mathematical Operations

- Assignment
  - X = X + Y : Add Y to X and store it in X; used as an accumulator
  - X = constant value : Store a constant value in X
- Calculation
  - +
  - –
  - *
  - /
  - **
  - …

OUTPUT

To an external entity: PRODUCE <name of data> TO <name of external entity>

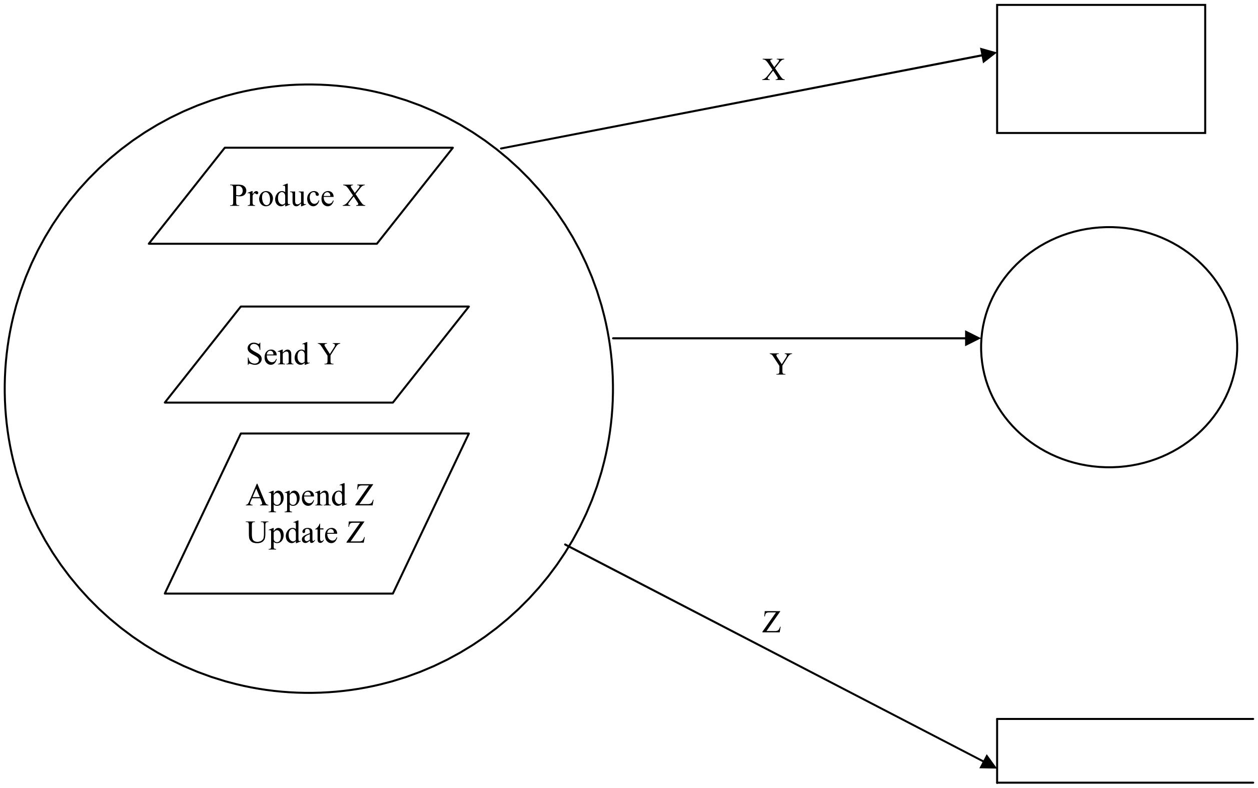
- PRODUCE account-balance TO customer

To another process: SEND <name of data> TO <name of process>

- SEND valid-order TO "Check Customer Credit"

To a data store:

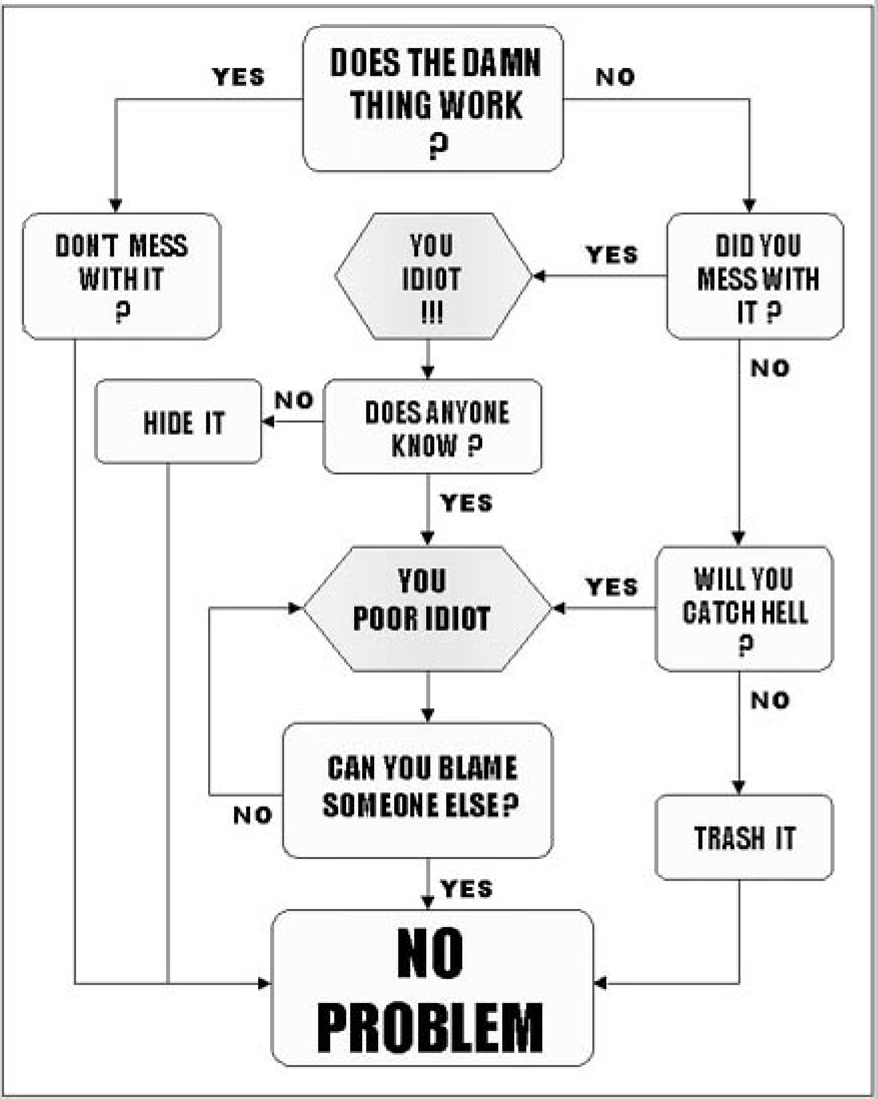
- Creating a new record: APPEND <name of data> TO <file name>
  - APPEND customer-details TO customer
- Updating an existing record: UPDATE <file name>: <field name(s)>
  - UPDATE inventory: quantity-on-hand



To declare the end of a program : END ○

* Remarks (to clarify the purpose or the workings of the subsequent code)

PROBLEM SOLVING FLOWCHART



Structured English Assignment: Payment Processing

Write structured English for the following process. The task being performed here is processing payments received from customers.

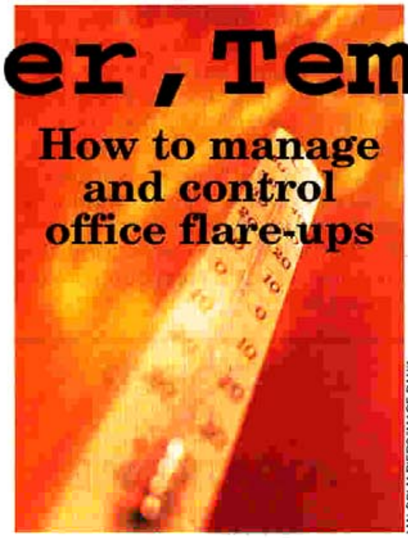
> *A payment is received from the customer. If it specifies customerid, the record of the customer is brought up from the CUSTOMER file. If such a record cannot be found, then the payment details generate a new record in the PAYMENT file with "unapplied cash" assigned to customerid. If a matching record can be found, then its current-balance is read, reduced by total-amount and updated accordingly, and a new payment record is generated as the payment details are written to the PAYMENT file. If the payment does not specify customerid, then the payment details generate a new record in the PAYMENT file with "unapplied cash" assigned to customerid.*

Draw a data flow diagram of the above process.

Structured English Assignment: Temper Control

According to the following article “Temper, Temper!”, how should you handle a criticism which is not valid and does not have a shred of truth to it, but you are afraid of confronting the critic?

Temper, Temper!



How to manage and control office flare-ups

MALCOLM PIERS/IMAGE BANK

BY TAMMY DARLING

The value of venting–yelling, shouting and otherwise thrusting those bottled-up emotions on the innocent and guilty alike–has basically gone the way of the "healthy" cigarette.

Recent studies indicate that losing control may actually be more harmful to our health than stifling anger. The *Journal of the American Medical Association*, for example, reported last year that men who experience outbursts of anger have twice the risk of stroke as men who control their tempers. A recent study at the University of Tennessee, U.S.A., found that women who vented their anger had less successful relationships and lower self-esteem.

Screaming at the boss, in any case, has never been a brilliant career move. So what's an employee to do with those pent-up workplace resentments? Too often, they rear their peevish heads as sarcasm, irritability, sloppy work or procrastination. They can also show up in the form of physical woes, such as headaches, back spasms and worse.

Unfortunately, few people know how to properly channel their anger. "The last thing we do is sit down in a productive way and work it out," observes William Wilmot, a professor of communications at the University of Montana, U.S.A., and coauthor of "Interpersonal Conflict" (McGraw Hill, 1997).

The first step is acknowledging that you are, in fact, angry, says Hendrie Weisinger, Ph.D., a psychologist, corporate educator and the author of "Emotional Intelligence at Work" (Jossey-Bass Publishers, 1997). "If you ignore your anger," she says, "it will swell until you blow up over something insignificant."

When you feel yourself getting upset or angry, take a time out. If your manager or coworker criticizes your work, don't react immediately. Step back and take an objective look at what was said. Could there be a ring of truth to it? If so, admit it and move on–nobody's perfect. Be sure you don't use a cooling-down period to brood over a train of angry thoughts.

If they're waiting for a reaction or comment, simply say, "I'm not sure about that. Let me get back to you." Never retaliate in kind. They may actually have a hidden agenda, such as trying to sabotage your career by provoking you. Or they may be bored, misguided or in a foul mood–or perhaps they just dislike you. The point is: Don't react. Listen.

It often helps to practice "feedback listening." Listen carefully when an argument is building. Notice the body language and the tone of voice of the other person, then repeat what you think you heard. Feedback listening promotes empathy and can even lead to the four magic words in conflict resolution: "You may be right."

STEVE MCALISTER/IMAGE BANK

Ask yourself what you're really upset about. Is it what was said, how it was said or the individual who said it? Ask yourself exactly what you hope to gain by speaking up. Sketch out a list of points you want to communicate.

When confronting the individual, remain calm and professional. When you start screaming, "You idiot!"–you've already lost the battle, observes Janice Walker Anderson, author of "Communication Skills for Surviving Conflicts at Work" (Hampton Press, 1996). You need to convey that you're upset or angry, but you should never appear irrational. Calmly and assertively explain your position and feelings. Don't become hostile or defensive. Stick to the issue at hand. And never bring up past grievances. According to Carol Scofield, who leads conflict management workshops in the United States and Canada: "When you bring in old baggage, you lose your credibility."

Convey understanding for the individual's viewpoint, but draw the line as to what you will and will not tolerate. Once you've made your point, stop and wait for a response. After you've dealt with the conflict, release your anger. Don't hold a grudge.

If you're afraid of confronting the person in such a manner, even a simple "point taken" may be all that's needed.

Ignoring a problem won't cause it to magically vanish, of course. But you can avoid constant power struggles by choosing your battles wisely. When someone pushes your buttons, don't get mad–work even harder. After all, success is the best revenge.

• *Tammy Darling practices self-control and free-lance writing in Three Springs, Pennsylvania, U.S.A.*

Why to Control Your Temper

- To prevent harmful effects to your health (stroke).
- To maintain your health.

- To prevent anger from swelling and blowing up later over something insignificant.
- To maintain your emotional stability.

- To prevent playing into the hands of adversaries who are trying to sabotage your career.
- To play the right politics.

How to Control Your Temper

Anger-Handling Module

```
DOWHILE (anger is building)
    Admit anger
    Do not react immediately
    Listen carefully
    Notice the body language and tone of voice of criticizer
    Repeat what you think you heard
    IF (a reaction is expected) THEN
        Say: “I’m not sure about that; let me get back to you.”
        Do not retaliate in kind
    ENDIF
ENDDO
```

Criticism-Handling Module

```
IF (criticism is valid) OR (if criticism has a ring of truth to it) THEN
    Admit it
    Say: “You may be right”
    Move on
    Do not brood over angry thoughts
ELSE (criticism is not valid and does not have a ring of truth to it)
    IF (not afraid of confronting the criticizer) THEN
        Determine what is to be gained by speaking up
        Make a list of points to be communicated
        DOWHILE (confronting the individual)
            Remain calm and professional
            Do not scream
            Assertively explain your position and feelings
            Do not become hostile or defensive
            Do not bring up past grievances (stick to the issues at hand)
            Convey understanding for the individual’s viewpoint
            Draw the line as to what you will/will not tolerate
        ENDDO
        Wait for a response
        Release your anger (don’t hold a grudge)
    ELSE
        Say: “point taken”
    ENDIF
ENDIF
```

Structured English Assignment: Averaging

Write structured English for the following process:

Input: a set of numbers entered by the user one by one until all numbers are entered
Output: the average of the given set of numbers.

Assume the agent in charge of executing your instructions (such as the computer) can perform *only* the following tasks:

i. for each designated variable, receive data one value at a time as input
ii. assign a value to a variable
iii. compare two variables and select one of two alternative actions
iv. perform arithmetic operations
v. repeat a group of actions
vi. output the resulting information

Structured English Assignment: Maximizing

Write structured English for the following process:

Input: a set of numbers entered by the user one by one until all numbers are entered
Output: the largest number in the set

Assume the agent in charge of executing your instructions (such as the computer) can perform *only* the following tasks:

i. for each designated variable, receive data one value at a time as input
ii. assign a value to a variable
iii. compare two variables and select one of two alternative actions
iv. perform arithmetic operations
v. repeat a group of actions
vi. output the resulting information

Structured English Assignment: Aging

Using **a single IF/THEN/ELSE structure**, write
(a) plain English logic
(b) structured English

for the following process:

Input: a date in the past (don't worry about verifying that it is indeed in the past and not future)
Output: the age (as of today when the program is run) of a person who was born on that date

Assume the agent in charge of executing your instructions (such as the computer) can perform *only* the following tasks:

i. for each designated variable, receive data one value at a time as input
ii. assign a value to a variable
iii. compare two variables and select one of two alternative actions
iv. perform arithmetic operations
v. repeat a group of actions
vi. output the resulting information

NOTE

The following functions are available to you:

- **DAY (date)**: returns the day of a date as a number; DAY (Feb 19, 2013) = 19. Special case: DAY (TODAY) returns the number corresponding to today's day of the month.
- **MONTH (date)**: returns the month of a date as a number; MONTH (Feb 19, 2013) = 2. Special case: MONTH (TODAY) returns the number corresponding to today's month of the year.
- **YEAR (date)**: returns the year of a date as a number; YEAR (Feb 19, 2013) = 2013. Special case: YEAR (TODAY) returns the number corresponding to today's year.

Remember to use **a single IF/THEN/ELSE structure.**

Structured English Assignment: Judging

In a certain contest, there is one judge, a number of candidates, and a number of criteria. The judge rates each candidate relative to each criterion on a 1-10 scale. Consider the process that receives the names of the candidates (one at a time) along with their ratings (one rating per criterion per candidate at a time), and determines/displays the winner – the candidate with the highest total rating. Ignore ties.

For this process:

(a) Write plain English logic

(b) Write structured English

(c) Perform desk-check on your structured English and document all desk-check details using the following data:

Criteria: Candidates:	C1	C2	C3
A	6	8	4
B	9	7	6
C	8	8	5

Assume the agent in charge of executing your instructions (such as the computer) can perform *only* the following tasks:

vii. for each designated variable, receive data one value at a time as input

viii. assign a value to a variable

ix. compare two variables and select one of two alternative actions

x. perform arithmetic operations

xi. repeat a group of actions

xii. output the resulting information

(use a separate or an additional page if necessary)

Structured English for Creating Structured English

Given: Problem Statement (Input/Output Specifications)

******************* Understand The Problem **********************

```
DOWHILE (The problem statement is not fully grasped)
        Identify and name all the output variables
        Identify and name all the input variables
ENDDO
```

********************** Find The Solution ************************

```
DOUNTIL MENTAL-OUTPUT = FORMAL-OUTPUT
        Design discovery data that are representative and typical
        Execute the process mentally based on the discovery data to produce the MENTAL-OUTPUT
        Identify the mental steps just taken to transform input into output ( = STEPS)
        Write down the STEPS in plain language
        Execute the STEPS formally to produce the FORMAL-OUTPUT
ENDDO
```

******************** Document The Solution **********************

```
DOUNTIL a formally correct PROGRAM is constructed
        Reformat the STEPS into PROGRAM in accordance with the syntax of structured English
ENDDO
```

*********************** Test/Debug The Solution ***********************

```
DOUNTIL MENTAL-OUTPUT = FORMAL-OUTPUT
        Design test data that are representative and typical, but different from discovery data
        Execute the process mentally based on the test data to produce the MENTAL-OUTPUT
        Execute the PROGRAM formally to produce the FORMAL-OUTPUT (= “desk check”)
        PERFORM "Understand The Problem"
        PERFORM "Find The Solution"
        PERFORM "Document The Solution"
ENDDO
```

The Process of Creating Structured English

Given: Problem Statement (Input/Output Specifications)

1. Understand The Problem

- Identify and name all the output variables
- Identify and name all the input variables

2. Find The Solution

- Design *discovery* data that are representative and typical
- Execute the process mentally based on the discovery data to produce the manual output
- Identify and write down the mental steps just taken

3. Document The Solution

- Reformat the mental steps in accordance with the syntax of structured English

4. Test/Debug The Solution

- Design *test* data that are representative and typical, but different from *discovery* data
- Execute the process mentally based on the test data to produce the manual output
- Execute the structured English to produce the program output (= "*desk check*")
- If manual output = program output ➔ SUCCESS!
- Otherwise repeat he entire process to see where things went wrong