Coding Tasks – Coding Structures

Variables

Copy the following pseudocode into our platform:

set a = 5 set b = 3 sum = a + b product = a * b output “The sum a and b is “, sum output “The product of a and b is “, product	***Explanation:*** ***Variable Declaration***: **a**, **b**, **sum**, and **product** are variables. ***Assignment:*** Values are assigned using **set**. ***Arithmetic Operations:*** **sum** stores the result of addition, and product stores the result of multiplication. ***Output:*** The computed values are displayed.
YOU TRY: 1. ***Modify the code to use different numbers for a and b and observe the results.*** 2. ***Extend the code to calculate and print the difference (a - b).*** 3. ***Modify the code to calculate the sum of the squares: ($a^2 + b^2$). Use the caret ^ for exponentiation.***	

User Input

output “Enter a value for a” store user input as a output “Enter a value for b” store user input as b sum = a + b product = a * b output “The sum a and b is “, sum output “The product of a and b is “, product	***Explanation:*** **store user input as <variable>**: Prompt the user to enter a value, then store it in the specified variable. **store input as <variable>:** Prompt the user to enter a value, then store it in the specified variable.
YOU TRY: 1. ***Modify the code to include a third variable c. Users input the value for variable c.*** 2. ***Include the variable c in the product and sum calculation.*** 3. ***Create a new variable, average, and calculate the average of the a, b, and c.***	

If…else… structure

output “Enter a value for a” store user input as a output “Enter a value for b” store user input as b if b = 0 output “You cannot divide by zero.” else quotient = a / b output “The quotient of a and b is “, quotient end	***Explanation:*** **if <condition>**: Start a conditional block. The block executes only if the condition evaluates to true. **else**: Define an alternate block to execute if the preceding **if** condition is false. **end**: Close an **if** or **else** block.
YOU TRY: 1. ***Write a conditional block to check if the fraction a / b is proper or improper and display the appropriate message.***	

Fixed Loop Structures

set m = 0 repeat loop 10 times output m m = m + 1 end loop	***Explanation:*** **repeat loop <N> times**: Execute the enclosed block exactly N times. **end loop**: Close a loop block.

YOU TRY:

1. ***Modify the code to count down from 10 to 1.***
2. ***Modify the code to output 10 consecutive even numbers***
3. ***Modify the code to output the multiplication table of a number, n. Create variable n.***
4. ***Modify the code to output the first 10 perfect squares.***
5. ***Modify the code to output the first 10 terms in the Fibonacci sequence.***

Conditional Loop Structures

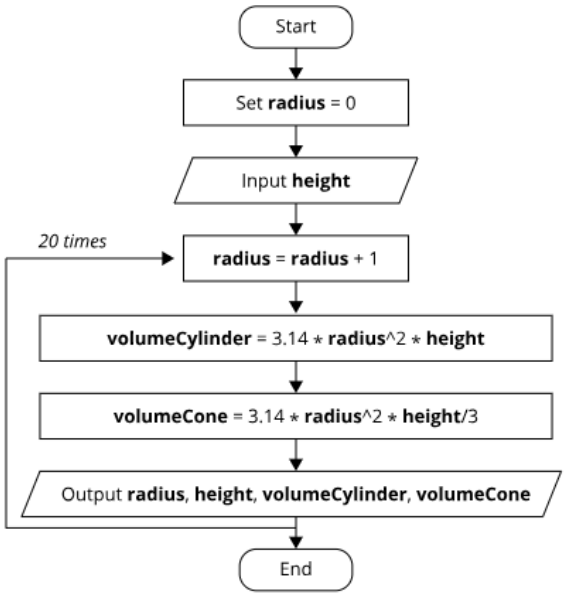
set m = 0 repeat loop until m > 20 output m m = m + 1 end loop	***Explanation:*** **repeat loop until <condition>**: Execute the enclosed block repeatedly until the condition becomes true.

YOU TRY:

1. ***Create variable n that stores a user inputted number. Modify the code to calculate the sum of the first n numbers.***

Coding Task: Volume Relationships

Read the flow chart and explain what it is modelling.



Explanation:

Let us turn the flow chart into pseudocode we can use to make our program.

```
set radius = 0
store user input as height
repeat loop 20 times
   radius = radius + 1
   volumeCylinder = 3.14 * (radius ^ 2) * height
   volumeCone = (3.14 * (radius ^ 2) * height) / 3
   output radius, "   ", height, "   ", volumeCylinder, "   ", volumeCone
end loop
```

Run the program and input a height value when prompted.

Observe the table of results that shows the radius, height, cylinder volume, and cone volume.

Questions

1. What happens to the volumes as the radius increases? Can you describe the relationship?

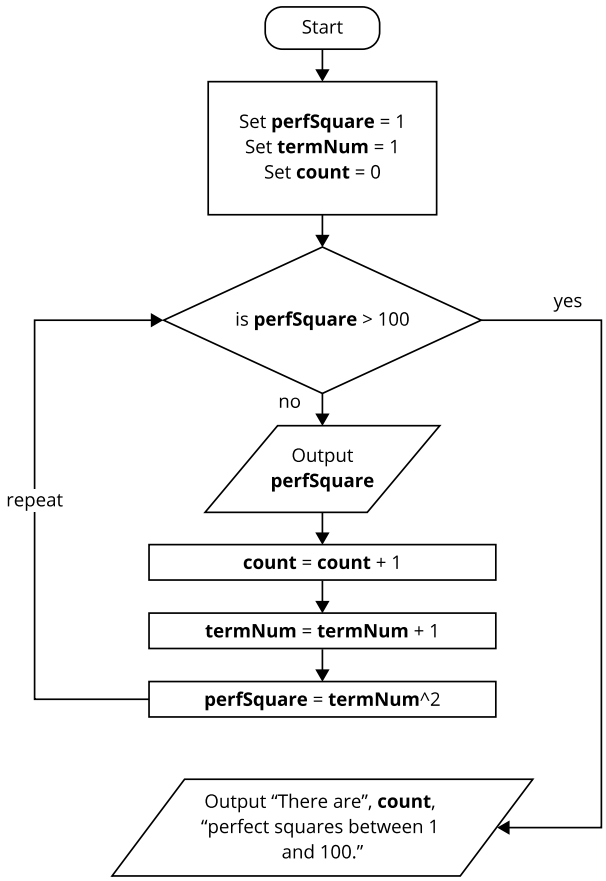
2. Modify the program to:
   - Include an option to input the radius range (e.g., from 5 to 25).
   - Calculate the surface area for both shapes.

Extension

Alter the code you created to compare the volumes and surface areas of prisms and pyramids when there is a change in one dimension, two dimensions, and three dimensions.

Coding Task: Perfect Squares

Explain how following the steps would generate the set of perfect squares from 1 to 100



Explanation:

Let us turn the flow chart into pseudocode we can use to make our program.

```
set perfSquare = 1
set termNum = 1
set count = 0
repeat loop until perfSquare > 100
output perfSquare
count = count + 1
termNum = termNum + 1
perfSquare = termNum ^ 2
end loop
output "There are ", count, " perfect squares between 1 and 100."
```

Run the program.

How many perfect squares are between 1 and 100?

Questions

1. Alter the code so that the program determines the perfect squares from 1 to 500, and from 1 to 1000. How many perfect squares are there?

2. Alter the code so that the program determines the perfect cubes from 1 to 100, and from 1 to 500. How many perfect squares are there?

Pseudocode Command Glossary

set <variable> = <value>: Define a variable and assign it a value.

store user input as <variable>: Prompt the user to enter a value, then store it in the specified variable.

store input as <variable>: Prompt the user to enter a value, then store it in the specified variable.

<variable> = input <text>: Prompt the user to enter a value, then store it in the specified variable.Text must be enclosed in double quotes ("").

output <text>, <variable>: Print a combination of text and variables. Text must be enclosed in double quotes ("").

display <text>, <variable>: Print a combination of text and variables. Text must be enclosed in double quotes ("").

if <condition>: Start a conditional block. The block executes only if the condition evaluates to true.

else: Define an alternate block to execute if the preceding **if** condition is false.

end: Close an **if** or **else** block.

repeat loop <N> times: Execute the enclosed block exactly N times.

repeat loop until <condition>: Execute the enclosed block repeatedly until the condition becomes true.

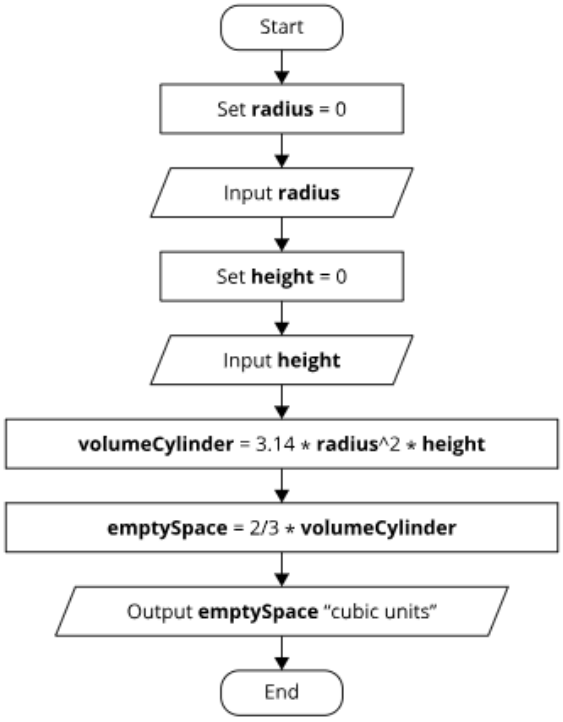
end loop: Close a loop block.

Arithmetic Expressions: Use standard arithmetic operators (`+`, `-`, `*`, `/`, `^`) in expressions to calculate values.

Exponentiation (^): Supports positive, negative, and decimal exponents. Example: `result = 4 ^ 0.5` evaluates to 2.

Predict the Outcome

Use the flow chart below. Use the glossary to create a pseudocode for this program.



What is the output? Use your program to test your answer.

Pseudocode Choice Board

Each square in the choice board below contains a pseudocode to modify and answer or a question that will require you to write your own pseudocode to answer. Complete _____ of the 6 choice boxes.

Question 1

```
set number = 0
output "Enter a number."
store user input as number
stepOne = number ^ 2
stepTwo = 3 * stepOne - 4
output stepTwo
```

1. What is the value of stepTwo if the input is 5?
2. What would the input have to be to get a value of 23 for stepTwo?
3. Modify the code so that if the input is 5, stepTwo will output 95. stepOne should remain unchanged.

```
stepTwo = _____ * stepOne - _____
```

4. Modify the original code so that when stepTwo is less than 10, you output the number, otherwise, you divide the number by 2 in a loop until stepTwo reaches below 10. You end the code and display a count of how many times the code divided by 2.

```
set number = 0
set count = ____
output "Enter a number."
store user input as number
stepOne = number ^ 2
stepTwo = 3 * stepOne - 4
repeat loop until _____________
stepTwo = stepTwo / _____________
count = _____________
end loop
output stepTwo
output "We divided by two ", _____________, " times. "
```

Question 2

```
variableB = 0
variableC = 0
output "Enter the value of variable B."
store input as variableB
output "Enter the value of variable C."
store input as variableC
variableA = ((5 * variableC) - variableB)/4
output "The value of variable A is ", variableA, "."
```

1. What is the value of variableA when the input for variableB is 1 and the input for variableC is 5?
2. What inputs could you use if you want the output of variableA to be 20?
3. Modify the code so that you calculate variableA five times by incrementing the input for variableB and variableC by 1 each time. Use a loop that repeats the calculation.

```
variableB = 0
variableC = 0
output "Enter the value of variable B."
store input as variableB
output "Enter the value of variable C."
store input as variable
repeat loop ___ times
variableA = ((5 * variableC) - variableB)/4
variableB = ___________ + ____
variableC = ___________ + ____
output "The value of variable A is ", variableA, "."
end ____
```

4. Modify the original code so that variableA only outputs when the sum of the inputs of variableB and variableC is greater than 10. Otherwise, output a message saying the sum if less than 10.

Question 3

```
amount = 0
interest = 0
output "Enter initial amount."
store user input as amount
output "Enter the simple interest rate as a percentage."
store user input as interest
finalAmount = amount + (amount * interest/100 * 5)
output "The final amount at the end of 5 years ", finalAmount, "."
```

1. How many different variables are in this line of pseudocode?

2. What is the final amount after 5 years if the initial amount is $100 and the interest rate is 6%

3. Modify the code for 10 years, 15 years, and 20 years. Track an investment of $1000 with an interest rate of 7% over these periods of time.

Years	Final Amount
5	
10	
15	
20	

4. Modify the code so that it can output the results like the table above, for every year from year 1 to 20.

Question 4

```
set amountOfLoan = 0.00
set borrowingTime = 0
set interestRate = 0.00
output "Enter the amount of the loan."
store user input as amountOfLoan
output "Enter the borrowing time for the loan in years."
store user input as borrowingTime
output "Enter the annual interest rate as a percentage."
store user input as interestRate
interestRate = interestRate/100
compoundPeriods = borrowingTime * 12
totalCost = amountOfLoan * (1 + interestRate/12)^compoundPeriods
totalInterest = totalCost - amountOfLoan
output "The total amount paid at the end of the loan would be ", totalCost
output "The total interest paid would be ", totalInterest
```

How would the total cost and total interest change if you adjust the compounding period to weekly? Semi-annually? Daily? Complete the chart.

Compounding Period	Total Cost	Total Interest
Daily		
Weekly		
Monthly		
Semi-Annually		
Annually		

Question 5

Write a pseudocode program to create a growing or shrinking pattern using a defined count. The following is an example of a program to create a shrinking pattern that generate the term numbers and values until the term value becomes less than or equal to −100. It also reports the number of positions in the pattern.

```
termValue = 200
termNum = 0
repeat loop until termValue < -100
termValue = termValue - 50
termNum = termNum + 1
output termNum
output termValue
end loop
output "There are ", termNum, " terms in this shrinking pattern."
```

Question 6

Write a pseudocode that will take the user input for x and y coordinates and checks to see if the coordinate satisfies the equation of a given line. Specify the slope and y-intercept of the line. Output if the point satisfies the equation of the line.