Hands-on Activity 3.1: Control Structures-Application

1.

```
Begin
// Initialize variables
  SET account_number
  SET beginning_balance
  SET total_charges
  SET total_credits
  SET credit_limit
  SET new_balance
// Input customer account details
  PRINT "Enter the account number:"
  INPUT account_number

  PRINT "Enter the balance at the beginning of the month:"
  INPUT beginning_balance

  PRINT "Enter the total of all items charged this month:"
  INPUT total_charges

  PRINT "Enter the total of all credits applied this month:"
  INPUT total_credits

  PRINT "Enter the allowed credit limit:"
  INPUT credit_limit
// Calculate new balance
     SET new_balance = beginning_balance + total_charges - total_credits
// Check if new balance exceeds credit limit
  IF new_balance > credit_limit THEN
    PRINT "Account Number:", account_number
    PRINT "Credit Limit:", credit_limit
    PRINT "New Balance:", new_balance
    PRINT "Credit limit exceeded."
ELSE
    PRINT "Account Number:", account_number
    PRINT "New Balance:", new_balance
    PRINT "Credit limit is not exceeded."
   END IF
END
```

2.

```
Begin
// Initialize variables
```

```
  SET total_miles = 0
  SET total_gallons = 0
  SET tankful_count = 0
// Input number of tankfuls
   PRINT "Enter the number of tankfuls:"
   INPUT tankful_count
// Loop for each tankful
   FOR i FROM 1 TO tankful_count DO
// Input miles driven and gallons used
   PRINT "Enter miles driven for tankful", i, ":"
   INPUT miles_driven

   PRINT "Enter gallons used for tankful", i, ":"
   INPUT gallons_used
// Calculate miles per gallon for this tankful
IF gallons_used > 0 THEN
SET mpg = miles_driven / gallons_used
   PRINT "Miles per gallon for tankful", i, "is", mpg
ELSE
   PRINT "Gallons used must be greater than zero. Please re-enter the values."
   DECREMENT i // Repeat the iteration for this tankful
   CONTINUE
END IF
// Update total miles and gallons
    total_miles = total_miles + miles_driven
    total_gallons = total_gallons + gallons_used
END FOR
// Calculate combined miles per gallon
IF total_gallons > 0 THEN
   SET combined_mpg = total_miles / total_gallons
   PRINT "Combined miles per gallon for all tankfuls is", combined_mpg
ELSE
   PRINT "Total gallons used must be greater than zero."
END IF
END
```

3.

```
Begin
// Declare variables
   SET weight AS FLOAT
   SET cost AS FLOAT
   SET extraWeightUnits AS INTEGER
// Input parcel weight
   PRINT "Enter parcel weight (max 1000g):"
```

```
   INPUT weight
// Determine shipping cost based on weight
IF weight <= 300 THEN
  SET cost = 5.00
ELSE IF weight <= 1000 THEN
// Calculate the number of 100g units above 300g, rounded up
  SET extraWeightUnits = (weight - 300) / 100
// Check if there's a remainder to round up
IF (weight - 300) MOD 100 > 0 THEN
   INCREMENT extraWeightUnits BY 1 // Round up
END IF
   SET cost = 5.00 + (extraWeightUnits * 2.00)
ELSE
   PRINT "Weight exceeds 1000g."
RETURN 1 // Exit program with an error code
END IF
// Output the calculated cost
   PRINT "Cost: P", cost FORMAT WITH 2 DECIMAL PLACES
RETURN 0 // Exit program successfully
END
```

4.

```
Begin
// Declare variables
   SET choice AS INTEGER
   SET value AS FLOAT
   SET result AS FLOAT
   SET repeat AS CHAR
DO
// Display conversion options
   PRINT "\n(1) cm - inches"
   PRINT "(2) inches - cm"
   PRINT "(3) feet - meters"
   PRINT "(4) meters - feet"
// Input user choice
   PRINT "Select (1-4):"
   INPUT choice
// Input value to convert
   PRINT "Enter value:"
   INPUT value
// Perform conversion based on user choice
   SWITCH choice DO
CASE 1:
result = value / 2.54
```

```
PRINT value, "cm = ", result, "inches"
BREAK

CASE 2:
result = value * 2.54
PRINT value, "inches = ", result, "cm"
BREAK

CASE 3:
result = value * 0.3048
PRINT value, "feet = ", result, "meters"
BREAK

CASE 4:
result = value / 0.3048
PRINT value, "meters = ", result, "feet"
BREAK

DEFAULT:
PRINT "Invalid choice"
END SWITCH

// Ask if the user wants to convert another value
PRINT "Convert another? (y/n):"
INPUT repeat

WHILE repeat == 'y'

RETURN 0
END
```

5.

```
Begin
// Declare variables
  SET choice AS INTEGER
  SET radius AS FLOAT
  SET length AS FLOAT
  SET width AS FLOAT
  SET base AS FLOAT
  SET height AS FLOAT
  SET side AS FLOAT
  SET area AS FLOAT
  SET repeat AS CHAR
```

```
DO
// Display area calculation options
PRINT "\n(1) Area of circle"
PRINT "(2) Area of rectangle"
PRINT "(3) Area of triangle"
PRINT "(4) Area of square"

// Input user choice
PRINT "Select (1-4):"
INPUT choice

// Calculate area based on user choice
SWITCH choice DO
CASE 1:
PRINT "Enter radius:"
INPUT radius
area = 3.1416 * radius * radius
PRINT "Area of circle: ", area FORMAT WITH 2 DECIMAL PLACES
BREAK
CASE 2:
PRINT "Enter length and width:"
INPUT length, width
area = length * width
PRINT "Area of rectangle: ", area FORMAT WITH 2 DECIMAL PLACES
BREAK
CASE 3:
PRINT "Enter base and height:"
INPUT base, height
area = 0.5 * base * height
PRINT "Area of triangle: ", area FORMAT WITH 2 DECIMAL PLACES
BREAK
CASE 4:
PRINT "Enter side:"
INPUT side
area = side * side
PRINT "Area of square: ", area FORMAT WITH 2 DECIMAL PLACES
BREAK
DEFAULT:
PRINT "Invalid choice"
END SWITCH
// Ask if the user wants to compute another area
PRINT "Compute another? (y/n):"
INPUT repeat
```

```
WHILE repeat == 'y'
RETURN 0
END
```