

```
#Class:      CSE1321L
#Section:    19
#Term:       Spring 19
#Instructor: Kevin Markley
#Name        Clay Waddell
#Lab#:       2
```

## Assignment 5

### **Problem 1-**

```
METHOD isValid(width, height):
  IF (width + height) > 30:
    RETURN True
  ELSE:
    RETURN False
END METHOD

METHOD my_area(width, height):
  IF isValid(width, height) is True:
    area = width * height
    RETURN area
  ELSE:
    RETURN False
END METHOD

METHOD my_perimeter(width, height):
  IF isValid(width, height) is True:
    perimeter = (width * 2) + (height * 2)
    RETURN perimeter
  ELSE:
    RETURN False
END METHOD
BEGIN MAIN
READ height
READ width
PRINT("Entered width: ", width, sep="")
PRINT("Entered height: ", height, sep="")

IF isValid(width, height) is False:
  PRINT("This is an invalid rectangle. Try again")
ELIF isValid(width, height):
  PRINT("Area: ", my_area(width, height), sep="")
  PRINT("Perimeter: ", my_perimeter(width, height), sep="")
END MAIN
```

## Problem 2 -

```
METHOD feet_meter(feet):
    meter = feet * 0.305
    RETURN meter
END METHOD

METHOD meter_feet(meter):
    feet = meter * 3.279
    RETURN feet
END METHOD

BEGIN MAIN

PRINT("Feet \t Meter \t\t Meter \t Feet")
FOR i in range(1, 21):
    PRINT(i, "\t\t", feet_meter(i), "\t\t", i, "\t\t", meter_feet(i), sep=")

END MAIN
```

## Problem 3 -

```
METHOD display_sums(i):
    x = 0
    PRINT("i", "\t", "Sum(i)")
    FOR j in range(1, (i + 1)):
        x += (j / (j + 1))
        PRINT(j, "\t", x)
    END METHOD

BEGIN MAIN
CALL display_sums(int(input("input integer: ")))
END MAIN
```

#### Problem 4 -

```
METHOD reverse(x):
    reverseNum = 0
    original = x
    while x > 0:
        y = x % 10
        reverseNum = (reverseNum * 10) + y
        x = x // 10
    answer = is_palindrome(reverseNum, original)
    RETURN answer
END METHOD
```

```
METHOD is_palindrome(x, y):
    IF x == y:
        RETURN True
    ELSE:
        RETURN False
END METHOD
```

```
METHOD is_prime(x):
    FOR i in range(2, x):
        IF (x % i) == 0:
            RETURN False
    RETURN True
END METHOD
```

BEGIN MAIN

```
i = 0
j = 1
counter = 0
```

```
WHILE i < 50:
    if reverse(j) is True and is_prime(j) is True:
        print(j, end='\t')
        i += 1
        counter += 1
        if counter % 10 == 0:
            print()
        j += 1
END MAIN
```

