

Lab 4P: Decision Structures and Boolean Logic_ 4P_DSBLX

GADDIS Text Pages 152 - 155

Using the IDLE editor (FILE/New Window).....

Name: _____ Jeff Couch _____

EXTRA CREDIT = 150 Pts (7 Projects)

For EACH Project you complete:

1. Create a **FLOWCHART** for each program and attach a **Screenshot(s)** of your **FLOWCHART**.
2. Take one **Screenshot** of both the EDIT window displaying your program and the SHELL window displaying your program results and paste as directed.

Project 2. **Areas of Rectangles** (15 Pts)

Input: Length1 = 30, Width1 = 45, Length2 = 25, Width2 = 50

2. Areas of Rectangles

The area of a rectangle is the rectangle's length times its width. Write a program that asks for the length and width of two rectangles. The program should tell the user which rectangle has the greater area, or if the areas are the same.

```

def getInfo():
    length = []
    width = []
    area = []

    for x in range(0, 2):
        length.insert(x, input("Input length of rectangle %d: " % int(x+1)))
        width.insert(x, input("Input width of rectangle %d: " % int(x+1)))
        area.insert(x, int(length[x])*int(width[x]))
    return area

def compareArea(area):
    if area[0] > area[1]:
        print("Rectangle 1 is larger than Rectangle 2.")
    elif area[0] < area[1]:
        print("Rectangle 2 is larger than Rectangle 1.")
    elif area[0] == area[1]:
        print("The rectangles are the same size.")

def main():
    compareArea(getInfo())

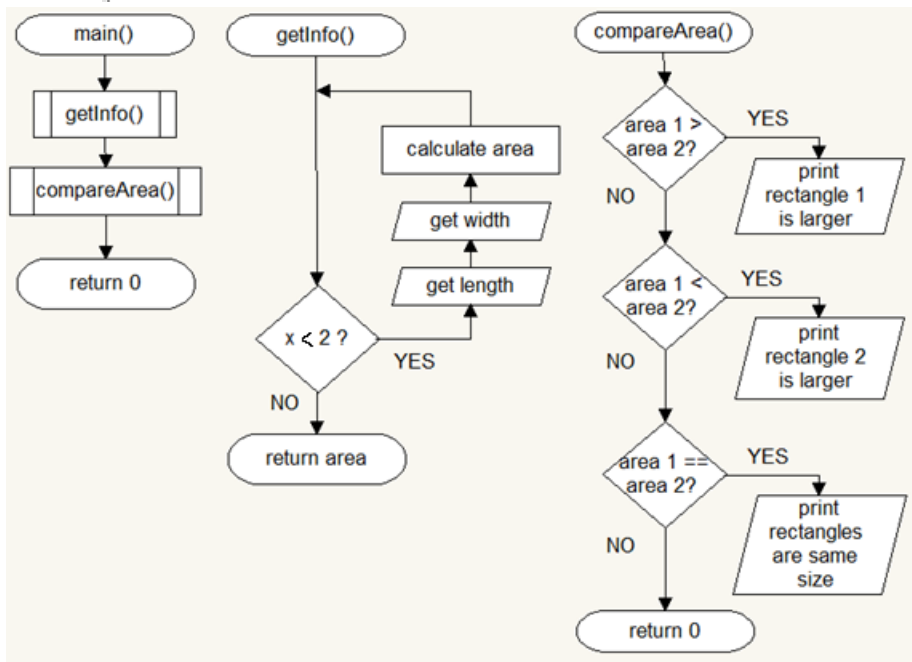
main()

```

```

>>>
Input length of rectangle 1: 30
Input width of rectangle 1: 45
Input length of rectangle 2: 25
Input width of rectangle 2: 50
Rectangle 1 is larger than Rectangle 2.
>>>

```



Project 3: Mass and Weight (20 Pts)

Input: mass(kilograms) = 225

3. Mass and Weight

Scientists measure an object's mass in kilograms and its weight in newtons. If you know the amount of mass of an object in kilograms, you can calculate its weight in newtons with the following formula:

$$\text{weight} = \text{mass} \times 9.8$$

Write a program that asks the user to enter an object's mass, and then calculates its weight. If the object weighs more than 1,000 newtons, display a message indicating that it is too heavy. If the object weighs less than 10 newtons, display a message indicating that it is too light.

```
def getMass():
    mass = float(input("Input object mass: "))
    return mass

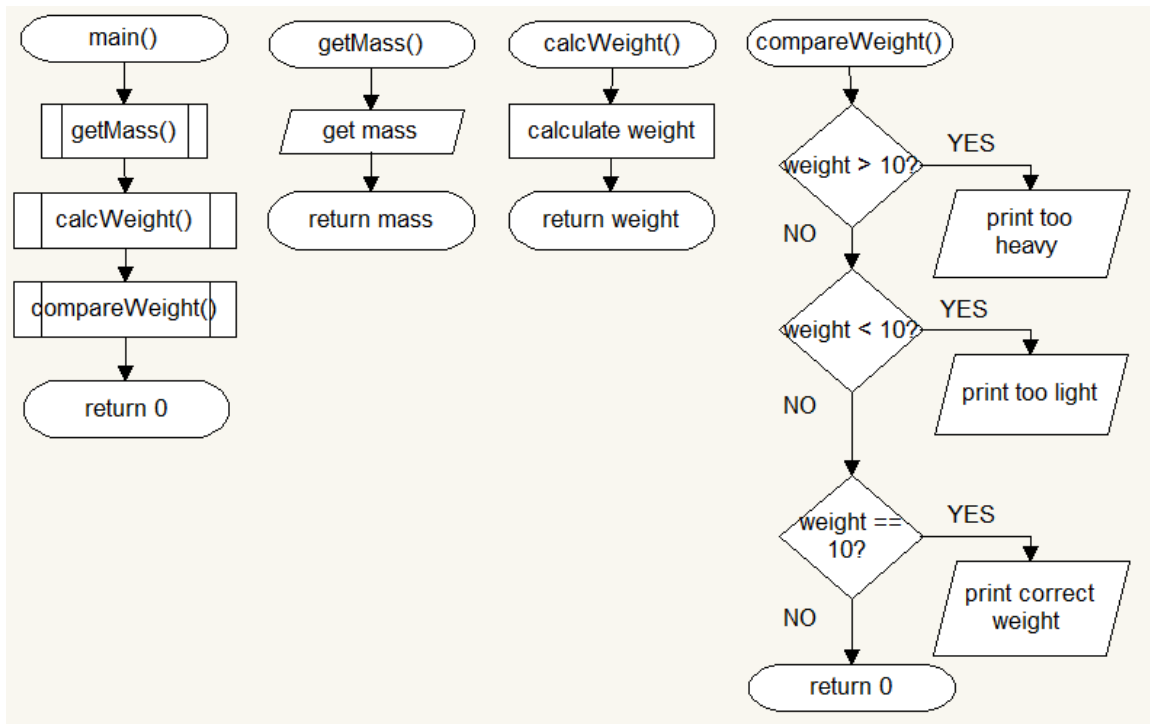
def calcWeight(mass):
    weight = mass * 9.8
    return weight

def compareWeight(weight):
    if weight > 10:
        print("The object is too heavy.")
    elif weight < 10:
        print("The object is too light.")
    elif weight == 10:
        print("The object is the correct weight.")

def main():
    compareWeight(calcWeight(getMass()))

main()

>>>
Input object mass: 225
The object is too heavy.
>>>
```



Project 5: Color Mixer (25 Pts)

Input: color1 = blue, color2 = red... color1 = green

5. Color Mixer

The colors red, blue, and yellow are known as the primary colors because they cannot be made by mixing other colors. When you mix two primary colors, you get a secondary color, as shown here:

- When you mix red and blue, you get purple.
- When you mix red and yellow, you get orange.
- When you mix blue and yellow, you get green.

Design a program that prompts the user to enter the names of two primary colors to mix. If the user enters anything other than "red," "blue," or "yellow," the program should display an error message. Otherwise, the program should display the name of the secondary color that results.

```

def getColors():
    colors = []
    for x in range(0, 2):
        colors.insert(x, input("Input primary color #{}: " % int(x+1)))
    return colors

def mixColors(colors):
    if colors[0] == 'red':
        if colors[1] == 'red':
            return 'red'
        if colors[1] == 'blue':
            return 'purple'
        if colors[1] == 'yellow':
            return 'orange'
        else:
            print("%s is not a primary color." % colors[1])

    if colors[0] == 'blue':
        if colors[1] == 'red':
            return 'purple'
        if colors[1] == 'blue':
            return 'blue'
        if colors[1] == 'yellow':
            return 'green'
        else:
            print("%s is not a primary color." % colors[1])


    if colors[0] == 'yellow':
        if colors[1] == 'red':
            return 'orange'
        if colors[1] == 'blue':
            return 'green'
        if colors[1] == 'yellow':
            return 'yellow'
        else:
            print("%s is not a primary color." % colors[1])

    else:
        print("%s is not a primary color." % colors[0])

def main():
    print("That makes %s." % mixColors(getColors()))

main()

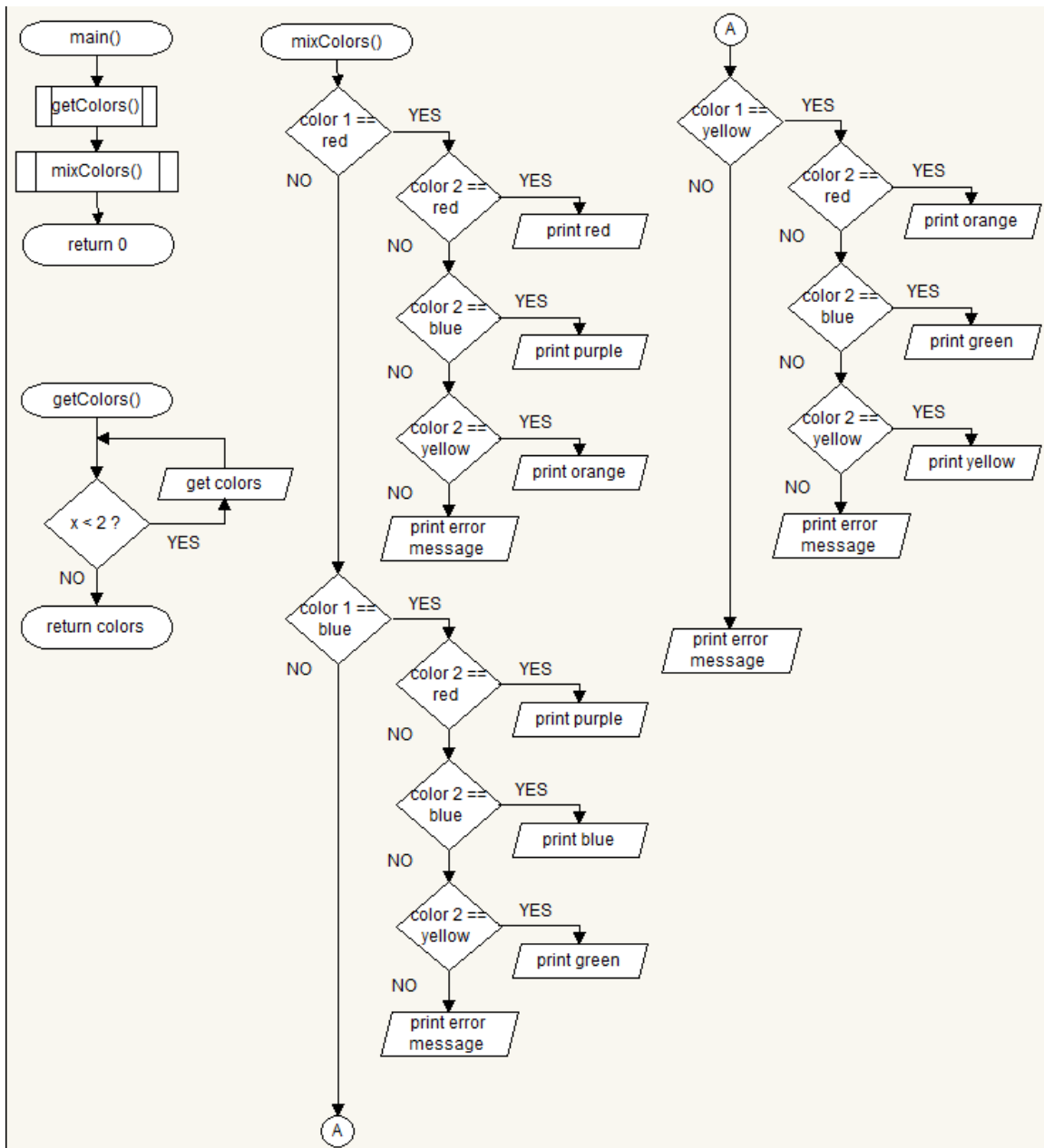
```



```

>>>
Input primary color #1: blue
Input primary color #2: red
That makes purple.
>>>

```



Project 6: **Change for a Dollar Game** (40 Pts)

6. **Change for a Dollar Game**

Create a change-counting game that gets the user to enter the number of coins required to make exactly one dollar. The program should prompt the user to enter the number of pennies, nickels, dimes, and quarters. If the total value of the coins entered is equal to one dollar, the program should congratulate the user for winning the game. Otherwise, the program should display a message indicating whether the amount entered was more than or less than one dollar.

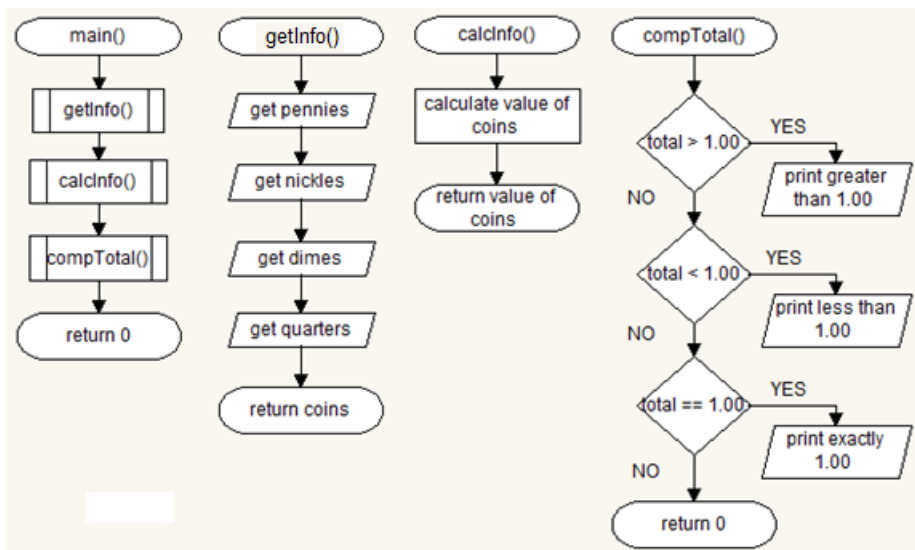
```
def getInfo():
    coins = {}
    coins['pennies'] = float(input("Number of pennies: "))
    coins['nickles'] = float(input("Number of nickles: "))
    coins['dimes'] = float(input("Number of dimes: "))
    coins['quarters'] = float(input("Number of quarters: "))
    return coins

def calcInfo(coins):
    pennyTotal = coins['pennies']*.01
    nickleTotal = coins['nickles']*.05
    dimeTotal = coins['dimes']*.10
    quarterTotal = coins['quarters']*.25
    return pennyTotal+nickleTotal+dimeTotal+quarterTotal

def compTotal(total):
    if total > 1.00 or total < 1.00:
        if total > 1.00:
            print("$%.2f is greater than $1.00." % total)
        if total < 1.00:
            print("$%.2f is less than $1.00." % total)
    elif total == 1.00:
        print("Congratulations! That is exactly $1.00!")
|
def main():
    compTotal(calcInfo(getInfo()))

main()

>>>
Number of pennies: 10
Number of nickles: 4
Number of dimes: 2
Number of quarters: 2
Congratulations! That is exactly $1.00!
>>>
```



Project 8: **Software Sales** (15 Pts)

Input: Packages = 25

8. Software Sales

A software company sells a package that retails for \$99. Quantity discounts are given according to the following table:

Quantity	Discount
10–19	20%
20–49	30%
50–99	40%
100 or more	50%

Write a program that asks the user to enter the number of packages purchased. The program should then display the amount of the discount (if any) and the total amount of the purchase after the discount.


```
def getInput():
    packages = int(input("Number of packages purchased: "))
    return packages

def discount(packages):
    if packages >= 10 and packages <= 19:
        discount = (packages*99) * .2
    elif packages >= 20 and packages <= 49:
        discount = (packages*99) * .3
    elif packages >= 50 and packages <= 99:
        discount = (packages*99) * .4
    elif packages >= 100:
        discount = (packages*99) * .5
    else:
        discount = 0

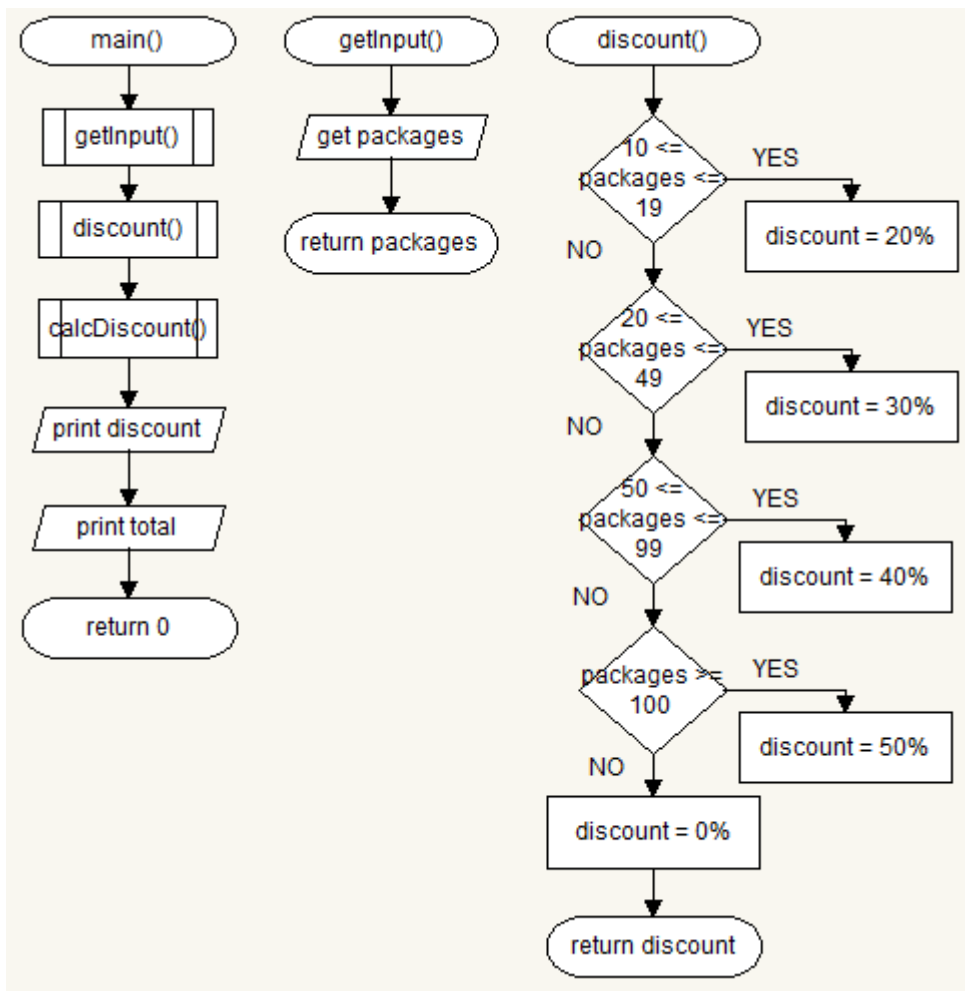
    return discount

def calcDiscount(discount, packages):
    subtotal = packages * 99
    total = subtotal - discount
    return total

def main():
    packages = getInput()
    print("Discount: $%.2f\nTotal: $%.2f" %
          (discount(packages), calcDiscount(discount(packages), packages)))
```

```
main()
```

```
>>>
Number of packages purchased: 25
Discount: $742.50
Total: $1732.50
>>>
```



Project 9: Shipping Charges (15 Pts)

Input: Weight(lbs) = 13

9. Shipping Charges

The Fast Freight Shipping Company charges the following rates:

Weight of Package	Rate per Pound
2 pounds or less	\$1.10
Over 2 pounds but not more than 6 pounds	\$2.20
Over 6 pounds but not more than 10 pounds	\$3.70
Over 10 pounds	\$3.80

Write a program that asks the user to enter the weight of a package and then displays the shipping charges.

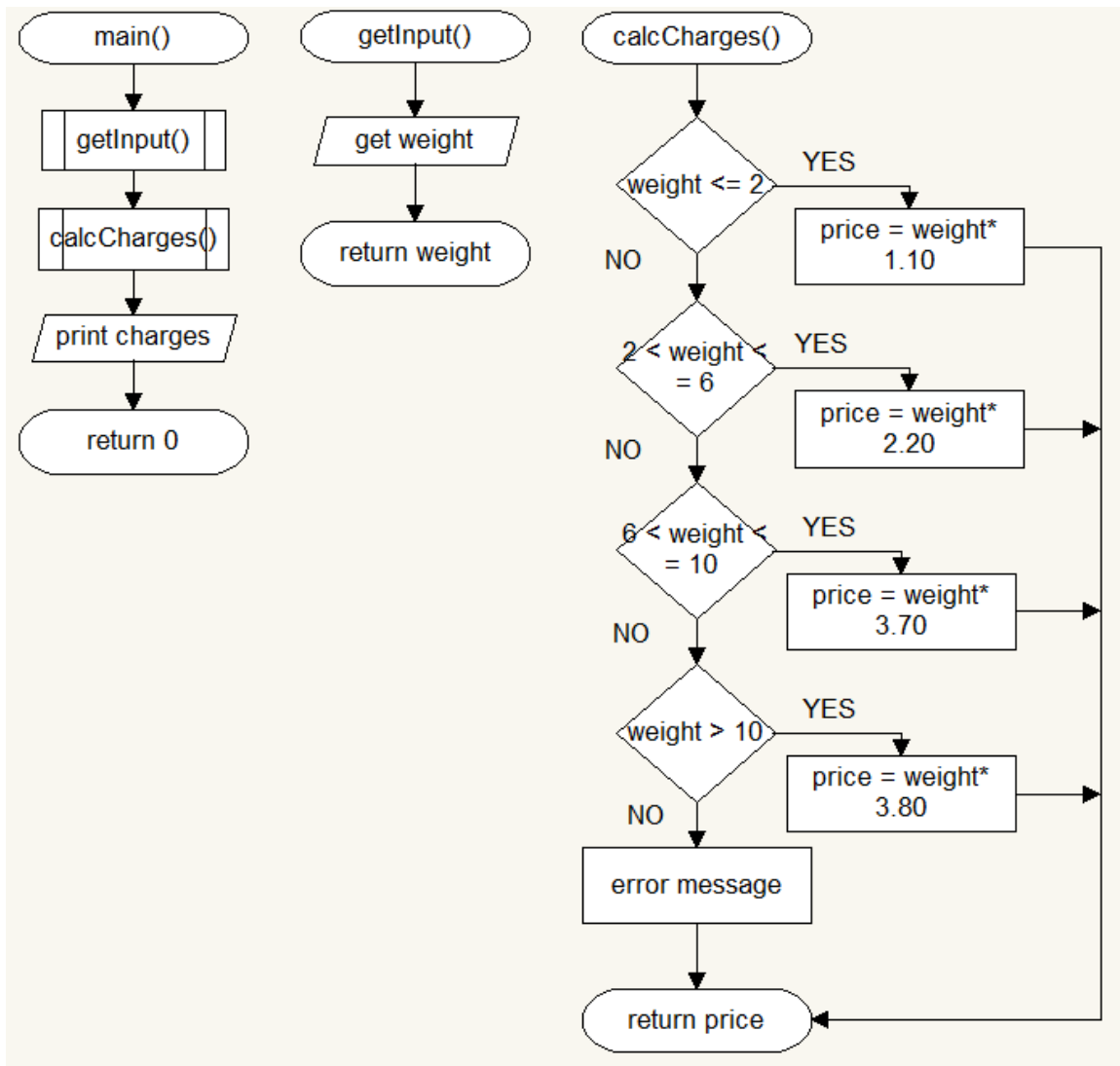
```
def getInput():
    weight = int(input("Weight of package: "))
    return weight

def calcCharges(weight):
    if weight <= 2:
        price = weight*1.10
    elif weight > 2 and weight <= 6:
        price = weight*2.20
    elif weight > 6 and weight <= 10:
        price = weight*3.70
    elif weight > 10:
        price = weight*3.80
    else:
        print("Invalid entry!")
        price = 0
    return price

def main():
    print("Shipping charges: $%.2f" % calcCharges(getInput()))

main()
```

```
>>>
Weight of package: 13
Shipping charges: $49.40
>>>
```



Project 10: Body Mass Index Program Enhancement (20 Pts)

Input: Weight(lbs) = 175, Height(In) = 69

10. Body Mass Index Program Enhancement

In programming Exercise #6 in Chapter 3 you were asked to write a program that calculates a person's body mass index (BMI). Recall from that exercise that the BMI is often used to determine whether a person is overweight or underweight for their height. A person's BMI is calculated with the formula

$$BMI = weight \times 703 / height^2$$

where *weight* is measured in pounds and *height* is measured in inches. Enhance the program so it displays a message indicating whether the person has optimal weight, is underweight, or is overweight. A person's weight is considered to be optimal if his or her BMI is between 18.5 and 25. If the BMI is less than 18.5, the person is considered to be underweight. If the BMI value is greater than 25, the person is considered to be overweight.

```

def getInfo():
    info = {}
    info['height'] = float(input("Height (in inches): "))
    info['weight'] = float(input("Weight (in pounds): "))
    return info

def calcBMI(height, weight):
    BMI = (weight*703/(height**2))
    return BMI

def optimal(BMI):
    if BMI < 18.5:
        print("You are underweight.")
    if BMI >= 18.5 and BMI <= 25:
        print("You are at optimal weight.")
    if BMI > 25:
        print("You are overweight.")

def main():
    info = getInfo()
    print("Your BMI is %.2f." % calcBMI(info['height'], info['weight']))
    optimal(calcBMI(info['height'], info['weight']))

main()

```

```

>>>
Height (in inches): 69
Weight (in pounds): 175
Your BMI is 25.84.
You are overweight.
>>>

```

