Java Foundations  
Practices - Section 6  
Problem 1: Validate a Bank PIN

Declare a valid integer PIN.  
 Prompt the user to enter the PIN.  
 In a while loop, perform the following steps:  
Compare the user-entered PIN with the already declared PIN  
 If the entered PIN is not the same, prompt the user to enter the PIN again  
 Repeat the loop until the correct PIN is entered  
 Print a message confirming that the correct PIN has been entered and that the user now has access to their account.  
The ValidatePin.java file is available to help you get started.

import java.util.Scanner;

public class ValidatePin {

public static void main(String[] args) {

// Declare a valid integer PIN

final int VALID\_PIN = 1234;

// Create a Scanner object to read user input

Scanner scanner = new Scanner(System.in);

// Variable to store the user's entered PIN

int enteredPin;

// Prompt the user to enter the PIN

System.out.print("Enter your PIN: ");

enteredPin = scanner.nextInt();

// While loop to check if the entered PIN matches the valid PIN

while (enteredPin != VALID\_PIN) {

System.out.println("Incorrect PIN. Please try again.");

System.out.print("Enter your PIN: ");

enteredPin = scanner.nextInt();

}

// Print confirmation message when the correct PIN is entered

System.out.println("Correct PIN entered. You now have access to your account.");

}

}

Problem 2: Displaying Multiples of a Number

Have the user enter a number, and then use a for loop to display all the multiples of that number from 1 to 12.

import java.util.Scanner;

public class Multiples {

public static void main(String[] args) {

// Create a Scanner object to read user input

Scanner scanner = new Scanner(System.in);

// Prompt the user to enter a number

System.out.print("Enter a number: ");

int number = scanner.nextInt();

// Use a for loop to display the multiples of the entered number from 1 to 12

for (int i = 1; i <= 12; i++) {

int multiple = number \* i;

System.out.println(number + " x " + i + " = " + multiple);

}

}

}

Problem 3: Programmatic ASCII Art

createRectangle(): This method accepts two arguments for width and height which should be used to print a rectangle  
 createTriangle(): This method accepts one argument for the size of a leg, which should be used to print an isosceles  
right triangle

public class LoopShape {

public static void createRectangle(int width, int height) {

if (width < 1 || height < 1) {

System.out.println("Invalid dimension. Please enter a value greater than 0.");

return;

}

for (int i = 0; i < height; i++) {

for (int j = 0; j < width; j++) {

System.out.print("#");

}

System.out.println();

}

}

public static void createTriangle(int leg) {

if (leg < 1) {

System.out.println("Invalid dimension. Please enter a value greater than 0.");

return;

}

for (int i = 0; i < leg; i++) {

for (int j = 0; j <= i; j++) {

System.out.print("#");

}

System.out.println();

}

}

}

And here is the LoopShapeTest.java file to test the methods:

public class LoopShapeTest {

public static void main(String[] args) {

LoopShape.createRectangle(5, 3);

System.out.println();

LoopShape.createTriangle(4);

System.out.println();

LoopShape.createRectangle(1, 1);

System.out.println();

LoopShape.createTriangle(1);

}

}