PIPER

Name: ____________________

Assessment

Guess My Number

1) What does the ask block do in the Pico program?
    a) It sets a variable to a random number.
    b) It retrieves a number from the user input in the console.
    c) It prints a message to the console.
    d) It compares two numbers.

2) Which block should you use to convert text to a number?
    a) Number from text
    b) Set Pico guess to
    c) Random integer from
    d) Print

3) What is the purpose of the repeat forever block in the game?
    a) To end the game once the player wins.
    b) To keep asking the player for a guess until they win.
    c) To display a message if the guess is correct.
    d) To generate a random number for the Pico.

4) How does the program determine if the player's guess is too high, too low, or correct?
    a) By comparing the player's guess with the Pico's guess using comparison blocks
    b) By generating a new random number for each guess
    c) By asking the player to guess again if the previous guess was incorrect
    d) By turning on the Pico's LED

5) What should you add to the code to ensure the game stops running once the player guesses correctly?
    a) A repeat forever block.
    b) An exit block after the "You won!" message.
    c) A print block that says "Too low."
    d) A set player guess to block.

6) How could you modify the game to make it more challenging, such as by changing the range of numbers or adding additional rules? Describe one or two modifications and how they would affect the game.

P I P E R

Name: ____________________

7) In real life, many security systems use codes or passwords to protect valuable items. How could the principles learned in this tutorial be applied to design a simple digital lock for a personal project, like a password-protected diary or a digital safe?

8) Look at the Python code you created. Right now, the game only gives the player a message if their guess is too high, too low, or correct. How could you change the code to give the player only 5 chances to guess the correct number? Write the new code below and explain what you changed.

PIPER

Name: ____________________

Answer Key
Guess My Number

1) B - It retrieves a number from the user input in the console.

2) A - Number from text

3) B - To keep asking the player for a guess until they win.

4) A - By comparing the player's guess with the Pico's guess using comparison blocks.

5) B - An exit block after the "You won!" message.

6) *Example:* Students might suggest increasing the range of numbers from 1 to 20 to a higher number, like 1 to 100, making the game more challenging as players would need to guess a number from a larger range. Another modification could be adding a limited number of guesses before the game ends, introducing a new difficulty level by requiring players to guess correctly within a set number of attempts.

7) *Example:* Students could apply the principles by using variables to store the correct password or code and implementing comparison blocks to check if the user's input matches the stored password. They could also add logic to provide feedback on whether the entered code is correct, too short, or too long. For example, in a digital diary, they could create a system where the diary prompts the user to enter a code, compares it to the stored code, and only grants access if the codes match.

8) Example:

```
import random
import time
import board
from piper_blockly import *

pico_guess = random.randint(1, 20)
attempts = 0
GP25 = piperPin(board.GP25, "GP25")

while attempts < 5:
    player_guess = float(input('Guess a number between 1 and 20: '))
    attempts += 1

    if player_guess < pico_guess:
        print('Too low')
    elif player_guess > pico_guess:
        print('Too high')
```

PIPER

Name: ___________________

```
    else:
        print('You won!')
        GP25.setPin(1)
        time.sleep(2)
        break

    time.sleep(0.5)

if attempts == 5 and player_guess != pico_guess:
    print("Sorry, you're out of guesses!")
```

Explanation:
The code now includes a variable called attempts that starts at 0. Each time the player makes a guess, attempts increases by 1. The while loop checks that the player still has guesses left. If the player reaches 5 attempts without guessing correctly, the game ends and prints a message letting the player know they're out of guesses.

This could also be done with a for loop, which can be used to limit the number of attempts to 5 as well.