1. What does DRY stand for and how do we accomplish that?

2. Functions can contain any code, we frequently write functions to carry out high-level and low-level tasks. What is the difference between these two types of tasks?

3. Well-designed functions should be SOFA, what does that mean?

4. What is Top-Down Program Design?
5. How and when should you test your code?

6. Two programmers wrote different versions of a game program, which is better?

```python
def play_again():
    answer = input("Play again? ")
    if answer == 'y':
        play_game()
        print("Well, fine!")

def play_game():
    secret = '13'
    num_guesses = 0
    while num_guesses < 3:
        guess = input("Guess: ")
        if guess == secret:
            num_guesses += 1
        else:
            print("You lose!")
    play_game()

play_game()
```

```python
def play_again():
    answer = input("Play again? ")
    return answer == y

def play_game():
    secret = '13'
    num_guesses = 0
    while num_guesses < 3:
        guess = input("Guess: ")
        if guess == secret:
            break
            num_guesses += 1
        if guess == secret:
            print("You win!")
        else:
            print("You lose!")
def play_games():
    keep_playing = True
    while keep_playing:
        play_game()
        keep_playing = play_again()
        print("Well, fine!")

play_games()
```
7. What functions should we write to program a computer version of Mastermind? Draw a diagram showing the program structure.

8. Write a function `count_exact` that takes in a guess and a secret code and counts the number of exact matches (correct color, correct place) – in other words, red pins.
9. Write a function `count_inexact` that takes in a guess and a secret code and counts the number of inexact matches (correct color, incorrect place) – in other words, white pins. Avoid double counting exact matches!

10. Write a function `is_valid` that takes in a guess, which is a string, and returns True if the guess is valid and False otherwise. A guess is valid if it contains 4 characters consisting only of the letters R, G, B, Y, P, and O.
11. Write a function `prompt_user` that repeatedly prompts the user for a guess until they enter a valid guess. This function should call `is_valid`. This function should not take any parameters and it should return a string, corresponding to a valid user guess.

12. Write a function `generate_code` that randomly generates a secret code. The secret code should be exactly four characters randomly selected (with replacement) from letters R, G, B, Y, P, and O.
13. Put all of these pieces together into a `play_game` function that plays a single game of mastermind. Then write a program that prompts the user to play mastermind and then re-prompts them to play again after the game ends.