



CODE CHALLENGE
EGYPT

PYTHON QUESTIONS

CODE CHALLENGE EGYPT

MID CATEGORY

20
23

Mission 1: Shapes 4 Fun

Overview:

Kids Empire, a popular kids' entertainment area located in Cairo, has been experiencing a decline in profitability over the past few months. The management has observed a shift in children's preferences towards interactive and technological games. So, to attract more children, Kids Empire has decided to hire you to design and implement a program that offers a simple interactive game called "Shapes 4 Fun".

Mission:

The program will provide a user-friendly interface that prompts children to enter the specifications of the shape they want to create. This includes details such as the shape type (**square, triangle, rectangle, or circle**), size (**small, medium, or large**), and color (**red, green, blue, orange, or yellow**) and based on these specifications, the program will utilize the turtle graphics library to draw the requested shape on the screen, allowing children to visually experience their creations.

As a bonus part, let the program be able to handle both lower and uppercase inputs for the same shape. For example, if the child wants to draw a square, all the following examples are considered as a square: **Square, square, SQUARE, SQuaRE** and the same goes for the other shapes.

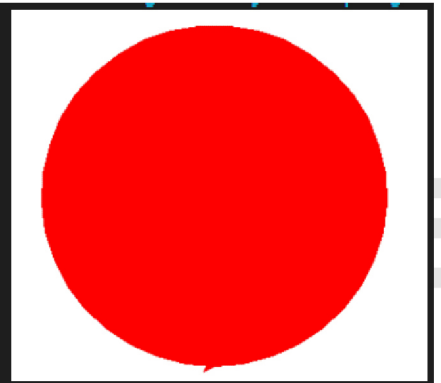
Useful Hints:

- ✓ Use **conditional statements** to handle different shape types and draw the corresponding shape based on user input.
- ✓ Utilize the turtle's **directions functions** to move the turtle and create the desired shape.
- ✓ Make a **user-defined function** for each shape to draw it based on the user input. Then, embed each one within its correct conditional statement.

Expected Output:

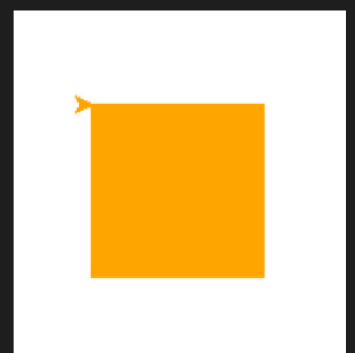
- First example → The shape's type is written in lower case letters

```
Welcome to the "Shapes 4 Fun" Interactive Game!  
Enter the shape (square, triangle, rectangle, circle): circle  
Enter the size (1 for small, 2 for medium, 3 for large): 3  
Enter the color (red, green, blue, orange, yellow): red
```



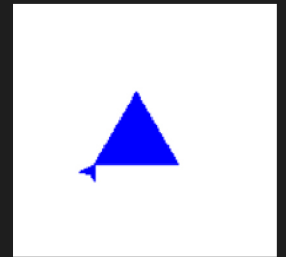
- Second example → The shape's type is written in upper case letters

```
Welcome to the "Shapes 4 Fun" Interactive Game!  
Enter the shape (square, triangle, rectangle, circle): SQUARE  
Enter the size (1 for small, 2 for medium, 3 for large): 2  
Enter the color (red, green, blue, orange, yellow): orange
```



- Third example → The shape's type is written as a capitalized word

```
Welcome to the "Shapes 4 Fun" Interactive Game!  
Enter the shape (square, triangle, rectangle, circle): Triangle  
Enter the size (1 for small, 2 for medium, 3 for large): 1  
Enter the color (red, green, blue, orange, yellow): blue  
|
```



CODE CHALLENGE

EGYPT

Mission 2: Guessing the Number

Overview:

The STEM-Tech Expo, an annual event dedicated to showcasing the latest advancements in science and technology, is gearing up for its next edition. To make the event's opening even more exciting, the organizers have decided to create a thrilling interactive game as one of the fun activities that attendees can play with through the event's opening or within breaks.

Mission:

The game idea is to take from the user 2 integers: the lower and upper limits of the game. Then, the program should generate a random number within that range and provide the user with a number of chances to guess the correct number based on that range before the chances are over. The game's logic goes as the following:

- 1- Welcoming the user and printing the game idea to the user to understand what should be done.
- 2- Asking the user to input the lower limit then the upper limit and generate the random number between those limits.
- 3- Calculating the number of chances that can be calculated by making: $\log_2(\text{the number of all possible numbers that the user can choose from})$. In simpler terms, that equals to: $\log_2(\text{upper limit} - \text{lower limit} + 1)$. Then, showing the user the number of available chances and asking to enter the first guess.

4- While the user has guesses left, compare the user's guess with the random number as the following:

- If the user's guess equals the random number → **Print congratulations to the user.**
- If the user's guess is smaller than the random number → **Print your guess is small, try bigger one.**
- If the user's guess is bigger than the random number → **Print your guess is big, try smaller one.**

5- If the user's chances are over, show to them the correct number and print better luck next time.

As a bonus part, make a variable named trial that shows the user the guess number when they enter their guess to keep track how many guesses are left. Additionally, show the trial's number which the user guessed the random number correctly.

Useful Hints:

- ✓ Use the **randint function from the random module** by passing the lower and upper limits inside it to generate the random number and store it in a variable.
- ✓ Use the **log2 and ceil functions from the math module** to calculate the number of chances that the user has to guess the correct number and to ceil the result to the nearest next integer respectively.
- ✓ Use a **while loop** to make the program keep taking the user's guesses until their guesses are over or **break the loop** if their guess is correct.

Expected Output:

Example of the whole program where lower and upper limits are 1, 10 respectively with the bonus part added.

```
Welcome to Guessing the number game!
You will give me 2 numbers
The lower and upper limits of the game.
Try to guess the number that will be generated randomly
Before your chances are over

Enter the lower limit: 1
Enter the upper limit: 10
You have only 4 chances to guess the correct number!
Enter guess 1: 5
Your guess is big, try smaller one!
Enter guess 2: 3
Your guess is big, try smaller one!
Enter guess 3: 2
Your guess is big, try smaller one!
Enter guess 4: 1
Congratulations!
You did it in trial 4
```

Example where lower and upper limits are 1, 20 respectively with the bonus part added.

```
Enter the lower limit: 1
Enter the upper limit: 20
You have only 5 chances to guess the correct number!
Enter guess 1: 10
Your guess is small, try bigger one!
Enter guess 2: 15
Your guess is small, try bigger one!
Enter guess 3: 18
Your guess is small, try bigger one!
Enter guess 4: 19
Your guess is small, try bigger one!
Enter guess 5: 20
Congratulations!
You did it in trial 5
PS F:\Data Science\Projects\Python projects> █
```

Example where lower and upper limits are 1, 100 respectively with the bonus part added.

```
Enter the lower limit: 1
Enter the upper limit: 100
You have only 7 chances to guess the correct number!
Enter guess 1: 60
Your guess is small, try bigger one!
Enter guess 2: 85
Your guess is big, try smaller one!
Enter guess 3: 70
Your guess is small, try bigger one!
Enter guess 4: 77
Congratulations!
You did it in trial 4
```

Example where lower and upper limits are 10, 50 respectively with the bonus part added but here, the user couldn't guess the correct number.

```
Enter the lower limit: 10
Enter the upper limit: 50
You have only 6 chances to guess the correct number!
Enter guess 1: 30
Your guess is big, try smaller one!
Enter guess 2: 20
Your guess is big, try smaller one!
Enter guess 3: 15
Your guess is big, try smaller one!
Enter guess 4: 10
Your guess is small, try bigger one!
Enter guess 5: 5
Your guess is small, try bigger one!
Enter guess 6: 12
Your guess is small, try bigger one!

The correct number is 14
Your guesses are over.
Better luck next time.
```

Mission 3: Pizza Ordering App

Overview:

A popular pizza restaurant is facing challenges in managing their orders efficiently. They currently use a manual system, resulting in frequent errors in order processing and customer dissatisfaction. So, you are tasked to develop a basic pizza ordering system that displays the menu, handles order additions, and calculates total costs. This system aims to streamline the ordering process, minimize errors, and enhance the overall customer experience.

Mission:

The system implementation will be using Object-Oriented Programming (OOP) principles and here are the steps that you should follow to accomplish that mission:

1- Create a class called 'Order' to manage the pizza ordering process that contains 4 methods as the following:

- `__init__(self)` → that initializes the 'Order' object. It sets up the initial state of the object, including the menu which is shown below and an empty list for order items.

```
self.menu = {  
    'beef': {'small': 80, 'medium': 95, 'large': 110},  
    'chicken': {'small': 75, 'medium': 90, 'large': 105},  
    'seafood': {'small': 110, 'medium': 135, 'large': 150},  
    'pepperoni': {'small': 70, 'medium': 90, 'large': 115}  
}
```


- `display_menu(self)` → that displays that menu to the user. It iterates through the menu dictionary and prints the available pizza options along with their sizes and prices.
- `add_pizza(self, pizza_name, size)` → that adds pizzas to the order. It also checks whether the provided pizza name and size are valid, and if so, it adds the pizza to the order list.
- `calculate_total_cost(self)` → that calculates the total cost of the order based on the items in the order list.

2- Create an object from the 'Order' class and call the `display_menu()` method to display the menu for the user before asking them to make their order.

3- Use a loop to interact with the user, asking them to input their desired pizzas and their corresponding sizes one by one until they type 'done'. After each order, display the order summary with final cost.

4- Handle the cases where the user enters an invalid pizza name or size. Ask them to re-enter a valid pizza and size names.

As a bonus part, ask the user if they want to place another order after showing the final cost in case, they want to add any other items. Then, include the new items ordered to display at the end all items the user ordered with the total cost.

Useful Hints:

- ✓ Use a **while loop** to continuously ask the user to enter their desired pizza name and its size until they type 'done'.
- ✓ Use the **items() method** to easily iterate over the dictionary keys and values when displaying the menu items to the user.
- ✓ Use the **formatted-string** to format and embed any variables easily inside the print() function.

Expected Output:

An example of typing 'done' first while the order list is empty

```
Welcome to our pizza app
Here are our pizza options:

beef pizza: small for 80 EGP, medium for 95 EGP, large for 110 EGP
chicken pizza: small for 75 EGP, medium for 90 EGP, large for 105 EGP
seafood pizza: small for 110 EGP, medium for 135 EGP, large for 150 EGP
pepperoni pizza: small for 70 EGP, medium for 90 EGP, large for 115 EGP
Enter pizza name or type "done" to finish your order: done
Your order is empty. Please enter your order.
Enter pizza name or type "done" to finish your order: chicken
Enter the size (small, medium, large): medium
Enter pizza name or type "done" to finish your order: done

Your order:
- medium chicken pizza
Total cost: 90 EGP
Would you like to place another order? (yes/no): no
Thank you for ordering from our pizza app!
```

An example of ordering beef and seafood pizzas without placing another order

```
Welcome to our pizza app
Here are our pizza options:

beef pizza: small for 80 EGP, medium for 95 EGP, large for 110 EGP
chicken pizza: small for 75 EGP, medium for 90 EGP, large for 105 EGP
seafood pizza: small for 110 EGP, medium for 135 EGP, large for 150 EGP
pepperoni pizza: small for 70 EGP, medium for 90 EGP, large for 115 EGP
Enter pizza name or type "done" to finish your order: beef
Enter the size (small, medium, large): large
Enter pizza name or type "done" to finish your order: seafood
Enter the size (small, medium, large): small
Enter pizza name or type "done" to finish your order: done

Your order:
- large beef pizza
- small seafood pizza
Total cost: 220 EGP
Would you like to place another order? (yes/no): no
Thank you for ordering from our pizza app!
```

An example of ordering invalid pizza name or pizza size

```
Welcome to our pizza app
Here are our pizza options:

beef pizza: small for 80 EGP, medium for 95 EGP, large for 110 EGP
chicken pizza: small for 75 EGP, medium for 90 EGP, large for 105 EGP
seafood pizza: small for 110 EGP, medium for 135 EGP, large for 150 EGP
pepperoni pizza: small for 70 EGP, medium for 90 EGP, large for 115 EGP
Enter pizza name or type "done" to finish your order: mix cheese
Enter the size (small, medium, large): large
Invalid pizza choice. Please choose from the menu options.
Enter pizza name or type "done" to finish your order: 
```

Invalid pizza name → mix cheese

Welcome to our pizza app
Here are our pizza options:

Invalid pizza size → family size

beef pizza: small for 80 EGP, medium for 95 EGP, large for 110 EGP
chicken pizza: small for 75 EGP, medium for 90 EGP, large for 105 EGP
seafood pizza: small for 110 EGP, medium for 135 EGP, large for 150 EGP
pepperoni pizza: small for 70 EGP, medium for 90 EGP, large for 115 EGP
Enter pizza name or type "done" to finish your order: pepperoni
Enter the size (small, medium, large): family
Invalid size. Please choose from small, medium, or large.
Enter pizza name or type "done" to finish your order: █

An example of ordering another order that will be added to the previous order

Here are our pizza options:

beef pizza: small for 80 EGP, medium for 95 EGP, large for 110 EGP
chicken pizza: small for 75 EGP, medium for 90 EGP, large for 105 EGP
seafood pizza: small for 110 EGP, medium for 135 EGP, large for 150 EGP
pepperoni pizza: small for 70 EGP, medium for 90 EGP, large for 115 EGP
Enter pizza name or type "done" to finish your order: chicken
Enter the size (small, medium, large): small
Enter pizza name or type "done" to finish your order: done

Your order:

- small chicken pizza

Total cost: 75 EGP

Would you like to place another order? (yes/no): yes

Enter pizza name or type "done" to finish your order: beef

Enter the size (small, medium, large): medium

Enter pizza name or type "done" to finish your order: done

Your order:

- small chicken pizza

- medium beef pizza

Total cost: 170 EGP

Would you like to place another order? (yes/no): no

Thank you for ordering from our pizza app!