2.1 Task 1: OOP Guess my number game

```python
import random
class GuessMyNumber:
    def __init__(self):
        self.number = random.randint(1, 100)
        self.guesses = 0

    def run(self):
        print("Computer chooses a number between 1 and 100")
        print("You will have 10 attempts to guess the correct number")

        while True:
            guess = input("Enter your guess: ")

            try:
                guess = int(guess)
            except ValueError:
                print("Invalid guess. Please enter another number.")
                continue

            self.guesses += 1

            if guess == self.number:
                print("Correct. Congratulations!".format(self.guesses))
                print("Good playing!")
                break
            elif guess < self.number:
                print("Incorrect. Guess Higher ")
            else:
                print("Incorrect. Guess Lower ")

if __name__ == "__main__":
    game = GuessMyNumber()
    game.run()
```

OUTPUT :

```
Computer chooses a number between 1 and 100
You will have 10 attempts to guess the correct number
Enter your guess: 50
Incorrect. Guess Higher
Enter your guess: 60
Incorrect. Guess Higher
Enter your guess: 70
Incorrect. Guess Higher
Enter your guess: 80
Incorrect. Guess Higher
Enter your guess: 90
Incorrect. Guess Lower
Enter your guess: 85
Incorrect. Guess Lower
Enter your guess: 84
Incorrect. Guess Lower
Enter your guess: 83
Incorrect. Guess Lower
Enter your guess: 82
Correct. Congratulations!
Good playing!
```

2.2 Task 2

```python
class Stack(object):
    def __init__(self, depth):
        self.depth = depth
        self.items = []

    def push(self, item):
        if len(self.items) < self.depth:
            self.items.append(item)
        else:
            raise OverflowError("Stack is full")

    def pop(self):
        if len(self.items) > 0:
            return self.items.pop()
        else:
            raise UnderflowError("Stack is empty")

    def is_empty(self):
        return len(self.items) == 0


# Test the class with several test cases
stack = Stack(5)
stack.push(1)
stack.push(2)
stack.push(3)
stack.push(4)
stack.push(5)

print(stack.pop())
print(stack.pop())
print(stack.pop())
print(stack.pop())
print(stack.pop())

print(stack.is_empty())
```

OUTPUT :

```
5
4
3
2
1
True
```