Problem 1: Simple ASCII Art

Overview Using text to create a picture is known as ASCII art. In section 2, we made an ASCII art cat. In this practice, you'll use print statements to recreate the image above. Task Use 8 print statements to recreate the smiley face above. Your art will rely on only a single character, besides space, such as X or #. The ProblemSet2_1 project is available to help you get started.

PROGRAM :-

```java
//N.V.Sandeep kumar
public class main {
    public static void main(String[] args) {
        System.out.println("  ....  ");
        System.out.println(" .    . ");
        System.out.println(". .  . .");
        System.out.println(".      .");
        System.out.println(". .  . .");
        System.out.println(" .    . ");
        System.out.println("  ....  ");
    }
}
```

OUTPUT :-

```
java -cp /tmp/9S7X5asMg1/main
  ....
 .    .
. .  . .
.      .
. .  . .
 .    .
  ....

=== Code Execution Successful ===
```

Problem 2: Original ASCII Art

Overview Using text to create a picture is known as ASCII art. In section 2, we made an ASCII art cat. In this practice, you'll create your own beautiful work of art. Task Use print statements to create your

own beautiful original ASCII art. Use comments to describe what your image is depicting. It's ok for your art to rely on only a single character, besides space, such as X or #. But you're encouraged to use a few different characters in your design, like in the cat example from class. Your art must also: • Use at least 8 print statements • Be at least 8 characters wide • Use at least 20 characters that aren't space You're welcome to create another cat. However, this image must look significantly different from the example used in class. Similarly, you're welcome to create another face, but it must look significantly different from the face in the previous practice (it's way too easy to turn the smiley face into a frowny face). Note: The backslash (\) character has special meaning in Java print statements. If you choose to use a backslash in your image, you'll actually need to write two backslashes (\\) in your print statement. The ProblemSet2_2 project is available to help you get started.

PROGRAM :-

```java
//N.V.Sandeep kumar
class Helloworld {
    public static void main(String[] args) {
        System.out.println("  /\\       /\\   ");
        System.out.println(" /  \\     /  \\ ");
        System.out.println("/      ---     \\ ");
        System.out.println("(   /\\      /\\   )");
        System.out.println("====    v    ===");
        System.out.println("====(_)|(_)===");
        System.out.println("   (_)|(_)=====");
        System.out.println("   (          ) ");
        System.out.println("    (_____)"  );
    }
}
```

OUTPUT :-

```
java -cp /tmp/yLiaUOW563/Helloworld
   /\        /\
  /  \      /  \
 /     ---      \
(   /\     /\   )
====    v    ===
====(_)|(_)===
   (_)|(_)=====
     (         )
      (_____)

=== Code Execution Successful ===
```

Problem 3:

The Snake Box Factory Overview Dear Respectable Software Engineer, Here at the world renowned Snake Box Factory, we pride ourselves on our ability to deliver the highest quality, custom sized, cardboard boxes to our customers. Our boxes are filled with the highest quality, custom-ordered snakes. We service thousands of accounts worldwide and have a solid 98% satisfaction rating with customers. However, the entire ordering process is currently written on cardboard, which is transported between departments via carrier snake. We thought this would be a good way to show confidence in the quality and usefulness of our product. But as our business continues to grow, we're realizing this was a bad idea. We believe it's time for a more conventional and digitized approach to our operations. Would you be able to help us develop the software we need to make this happen? Sincerely, President George Johnson, The Snake Box Factory

Tasks

Read the scenario found in the overview and consider what objects could be modeled as part of creating a software solution. Identify 3 objects from this scenario (remember, objects can be either tangible or abstract. List 3 properties and 3 behaviors belonging to each object.

Write your solution as a document rather than a .java file

PROGRAM :-

```java
//N.V.Sandeep kumar
class Customer {
    private String name;
    private int accountNumber;
    private double satisfactionRating;
    public Customer(String name, int accountNumber, double
        satisfactionRating) {
        this.name = name;
        this.accountNumber = accountNumber;
        this.satisfactionRating = satisfactionRating;
    }
    public void placeOrder(Order order) {
        System.out.println(name + " placed an order with ID: " + order
            .getOrderId());
    }
    public void trackOrder(Order order) {
        System.out.println("Order " + order.getOrderId() + " is
            currently " + order.getStatus());
    }
    public void provideFeedback(double rating) {
        this.satisfactionRating = rating;
        System.out.println(name + " provided feedback. New
            satisfaction rating: " + rating);
    }
}
```

```java
// Order class
class Order {
    private int orderId;
    private String boxSize;
    private String snakeType;
    private String status;
    public Order(int orderId, String boxSize, String snakeType) {
        this.orderId = orderId;
        this.boxSize = boxSize;
        this.snakeType = snakeType;
        this.status = "Processing";
    }
    public int getOrderId() {
        return orderId;
    }
    public String getStatus() {
        return status;
    }
    public double calculateCost() {
        // Simplified cost calculation
        return boxSize.length() * 10 + snakeType.length() * 20;
    }
    public void updateStatus(String status) {
        this.status = status;
        System.out.println("Order " + orderId + " status updated to "
            + status);
    }
    public void generateInvoice() {
        System.out.println("Invoice for Order " + orderId + ":");
```

```java
            System.out.println("Box Size: " + boxSize);
            System.out.println("Snake Type: " + snakeType);
            System.out.println("Total Cost: $" + calculateCost());
        }
    }
    // CarrierSnake class
class CarrierSnake {
        private int id;
        private double speed;
        private int capacity;
        public CarrierSnake(int id, double speed, int capacity) {
            this.id = id;
            this.speed = speed;
            this.capacity = capacity;
        }
        public void transportOrder(Order order) {
            System.out.println("Carrier Snake " + id + " is transporting
                order " + order.getOrderId());
        }
        public boolean checkAvailability() {
            // Simplified availability check
            return true;
        }
        public void reportLocation() {
            // Simplified location reporting
            System.out.println("Carrier Snake " + id + " is at the
                dispatch center.");
        }
    }
```

```java
// Main class to demonstrate the interactions
public class SnakeBoxFactory {
    public static void main(String[] args) {
        Customer customer = new Customer("Alice", 1001, 4.5);
        Order order = new Order(2001, "Large", "Python");
        CarrierSnake carrierSnake = new CarrierSnake(1, 10.5, 5);
        customer.placeOrder(order);
        order.generateInvoice();
        carrierSnake.transportOrder(order);
        carrierSnake.reportLocation();
        order.updateStatus("Shipped");
        customer.trackOrder(order);
        customer.provideFeedback(4.8);
    }
}
```

OUTPUT :-

```
java -cp /tmp/MoIk5MYvGD/SnakeBoxFactory
Alice placed an order with ID: 2001
Invoice for Order 2001:
Box Size: Large
Snake Type: Python
Total Cost: $170.0
Carrier Snake 1 is transporting order 2001
Carrier Snake 1 is at the dispatch center.
Order 2001 status updated to Shipped
Order 2001 is currently Shipped
Alice provided feedback. New satisfaction rating: 4.8

=== Code Execution Successful ===
```