Single-Level-Inheritance :--

Question 1 (Zepto – Grocery Delivery Parent Class: Order)

• Instance variables: order_id, items, amount (initialize inside __init__)

• Method:

show_order() → ▪ Create a local variable tax = 50

▪ Print all order details + tax Child Class: Delivery • Method: o
show_delivery() → ▪ First, call parent method show_order()

▪ Then print delivery status Object Creation

• Create an object of Delivery with all details.

• Call show_delivery() → this will show both order (parent) + delivery (child).

Ans:

```
class order_details:
    def __init__(self,order_id,items,amount):
        self.order_id= order_id
        self.items= items
        self.amount=amount
    def show_order(self):
        tax = 50
        print(f"Order ID is :{self.order_id},No.of items: {self.items}, {self.amount} + {tax} ")
class child(order_details):
    def show_delivery(self):
        super().show_order()
        self.delivery_status="delivered"
        print(f"Your order is:{self.delivery_status}")
obj = child(1,"Maggi",50)
obj.show_delivery()
obj1 = child(2,"Yipee",40)
obj1.show_delivery()
```

OUTPUT:

```
Order ID is :1,No.of items: Maggi, 50 + 50

Order ID is :2,No.of items: Yipee, 40 + 50
```

Question 2 (Amazon – E-commerce Parent Class: Product)

• Instance variables: name, price, category

• Method:

show_product() → ▪ Create a local variable platform = "Amazon"

▪ Print product details + platform Child Class: DiscountedProduct

• Method:

show_discount(discount_percentage) →

▪ Call show_product() from parent

▪ Create a local variable final_price = price - (price * discount_percentage / 100)

▪ Print discount details + final price

Object Creation

• Create an object of DiscountedProduct.

• Call show_discount() → this will show product details + discounted price.

ANS:

```
class product:
    def __init__(self,name,price,category):
        self.name = name
        self.price = price
        self.category = category
    def show_product(self):
        platform = "amazon"
        print(f"Details of Product:---
Name:{self.name},Price:{self.price},category:{self.category},{platform}")
class DiscountedProduct(product):
    def show_discount(self,discount_percentage):
        self.discount_percentage = discount_percentage
        super().show_product()
        final_price = self.price - (self.price * discount_percentage / 100)
        print(f" {final_price}")
obj = DiscountedProduct("kettle",500,"Appliances")
obj.show_discount(50)
```

OUTPUT:

```
Details of Product:--- Name:kettle,Price:500,category:Appliances,amazon
 250.0
```

Question 3 (Uber – Ride Booking Parent Class: Ride)

- Instance variables: ride_id, pickup, drop
- Method:

show_ride() →

  - Create a local variable distance = 12
  - Print ride details + distance Child Class: Driver

- Method: show_driver() →
  - Call show_ride() from parent
  - Print driver status Object Creation
- Create an object of Driver.
- Call show_driver() → this will show ride (parent) + driver (child).

ANS:

```
class ride:
    def __init__(self,ride_id,pickup,drop):
        self.ride_id = ride_id
        self.pickup = pickup
        self.drop = drop
    def show_ride(self):
        distance = 12
        print(f"Ride details:\n Ride_id:{self.ride_id},Pickup: {self.pickup}, Drop:{self.drop}, Distance: {distance}")
class driver(ride):
    def show_driver(self):
        super().show_ride()
        print("Driver status: Reaching The Destination")
obj = driver(19,"KPHB","MG Busstand")
obj.show_driver()
```

OUTPUT:

```
Ride details:
 Ride_id:19,Pickup: KPHB, Drop:MG Busstand, Distance: 12
Driver status: Reaching The Destination
```