Exception Handling Problem Statements

1. Online Payment Processing System

- **Problem Statement:** Create a function to simulate an online payment system that processes payments based on user input. Handle exceptions such as invalid card numbers (not exactly 16 digits), insufficient funds (payment amount greater than balance), and network errors (simulate using a random failure).
- **Sample Input:**

```
process_payment(card_number="1234567890123456", amount=100, balance=50)
```

- **Sample Output:**

```
Error: Insufficient funds. Available balance: 50.
```

- **Handling Exceptions:**
    - ValueError for invalid card number length.
    - Custom exception for insufficient funds.
    - Simulated network error using random failure.

2. Email Automation and Notification Service

- **Problem Statement:** Write a function that sends email notifications to a list of email addresses. Handle exceptions such as invalid email format (missing @), server connection errors, and failed sends.
- **Sample Input:**

```
send_email(to_address="invalid_email.com", subject="Reminder", body="This is a test.")
```

- **Sample Output:**

```
Error: Invalid email address format: invalid_email.com
```

- **Handling Exceptions:**
    - ValueError for invalid email format.
    - Custom exception for server connection errors.
    - Handle failure with retries and logging errors.

3. Web Scraping and Data Extraction Tool

- **Problem Statement:** Create a script that scrapes data from a webpage and writes it to a file. Handle exceptions such as HTTP errors, missing elements, and invalid URLs.
- **Sample Input:**

```
scrape_data(url="https://invalidurl.com")
```

- **Sample Output:**

```
Error: Invalid URL or page not found.
```

- **Handling Exceptions:**
  - Use `try-except` to handle `requests` library exceptions.
  - Handle missing elements with `KeyError`.
  - Log errors for further debugging.

4. File Backup and Recovery System

- **Problem Statement:** Write a script to copy files from a source to a backup location. Handle exceptions such as file not found, permission errors, and insufficient disk space.
- **Sample Input:**

```
backup_file(source="data.txt", destination="/backup/data.txt")
```

- **Sample Output:**

```
Error: File not found: data.txt
```

- **Handling Exceptions:**
  - FileNotFoundError for missing source files.
  - PermissionError for access issues.
  - Handle disk space issues with custom error messages.

5. **IoT Device Monitoring and Control System**

- **Problem Statement:** Create a function to simulate controlling IoT devices by sending commands. Handle exceptions such as device not responding, command timeouts, and invalid commands.
- **Sample Input:**

```
control_device(device_id=1, command="TURN_ON")
```

- **Sample Output:**

```
Error: Device not responding. Please check the connection.
```

- **Handling Exceptions:**
  - TimeoutError for device response delays.
  - ValueError for invalid command formats.
  - Custom error handling for device disconnections.

---

6. **Stock Trading Bot**

- **Problem Statement:** Create a function that simulates buying and selling stocks. Handle exceptions such as invalid stock symbols, insufficient funds, and network errors.
- **Sample Input:**

```
trade_stock(symbol="XYZ", action="buy", amount=100, balance=50)
```

- **Sample Output:**

```
Error: Insufficient funds. Available balance: 50.
```

- **Handling Exceptions:**
  - ValueError for invalid stock symbols.
  - Custom exception for insufficient funds.
  - Handle network issues with retries and logging.

---

7. **Data Import and Validation for Database**

- **Problem Statement:** Write a function that imports data from a CSV file into a system. Handle exceptions such as missing files, invalid data types, and database connection errors.
- **Sample Input:**

```
import_data(file_path="data.csv")
```

- **Sample Output:**

```
Error: File not found: data.csv
```

- **Handling Exceptions:**
    - FileNotFoundError for missing CSV files.
    - ValueError for invalid data types in CSV.
    - Handle database connection errors with appropriate messaging.

8. **Real-Time Weather Data Application**

- **Problem Statement:** Develop a function that fetches weather data from an API and displays it. Handle exceptions such as API connection errors, invalid API keys, and rate limit exceeded.
- **Sample Input:**

```
get_weather_data(api_key="invalid_key", location="New York")
```

- **Sample Output:**

```
Error: Invalid API key. Please check your credentials.
```

- **Handling Exceptions:**
    - ConnectionError for network issues.
    - KeyError for missing data fields in API response.
    - Custom exception for invalid API key handling.

9. **Image Processing and Analysis Tool**

- **Problem Statement:** Create a function to process images by applying filters. Handle exceptions such as unsupported file formats, file not found, and memory errors for large files.
- **Sample Input:**

```
process_image(file_path="image.bmp", filter_type="blur")
```

- **Sample Output:**

```
Error: Unsupported file format: image.bmp
```

- **Handling Exceptions:**
  - FileNotFoundError for missing image files.
  - TypeError for unsupported image formats.
  - MemoryError for large files.

---

10. **Taxi Booking and Dispatch System**

- **Problem Statement:** Write a function that simulates booking a taxi. Handle exceptions such as GPS errors, unavailable drivers, and payment processing errors.
- **Sample Input:**

```
book_taxi(pickup_location="Unknown", dropoff_location="City Center")
```

- **Sample Output:**

```
Error: GPS location not found. Please enter a valid pickup location.
```

- **Handling Exceptions:**
  - ValueError for invalid location inputs.
  - Custom exception for driver unavailability.
  - Handle payment errors with retries and logging.