মেট্রোপলিটন ইউনিভার্সিটি
Metropolitan
University

Department of Computer Science and Engineering

Course Code: CSE222	Credits: 1.5
Course Name: Object Oriented Programming Lab	Faculty: FRS

Lab 03 – Inheritance and Method Overriding

Task 1: Create a class hierarchy to model different types of vehicles.

- Start with a base class called "Vehicle" that has attributes "makeBy" and "makingYear", along with a method to display vehicle information.
- Add a method in the Vehicle class to check if the manufacturing year is a palindrome.
- Then, create two subclasses "Car" and "Motorcycle" that inherit from the Vehicle class.
- Add specific attributes and methods for each subclass.
- Finally, create instances of both Car and Motorcycle classes, and demonstrate the inheritance and method overriding.

Output:

```
Car Information:
Make: Toyota
Year: 2002
Number of doors: 4
This Car manufacturing year is a palindrome.

Motorcycle Information:
Make: Harley-Davidson
Year: 2003
Has sidecar: true
The Motorcycle manufacturing year is not a palindrome.
```

Task 2: Create a class hierarchy to model different types of animals.

- Start with a base class called "Animal" that has attributes "name" and "age", along with a method to make the animal sound.
- Add a method in the Animal class to find the factorial of the age of an animal.
- Then, create two subclasses "Dog" and "Cat" that inherit from the Animal class.
- Add specific attributes and methods for each subclass.
- Finally, create instances of both Dog and Cat classes and demonstrate the inheritance and method overriding.

Output:

```
Dog Name: Iron
Age: 7
The dog barks.
The factorial of 7 is 5040

Cat Name: Tusk
Age: 5
The cat meows.
The factorial of 5 is 120
```

Task 3: Create a class hierarchy for geometric shapes.

- Start with a base class called "Shape" that has an abstract method to calculate area.
- Then, create two subclasses "Rectangle" and "Triangle" that inherit from the Shape class.
- Implement the area calculation methods for each subclass.
- Add a method in the Triangle class to determine if it's a right-angled triangle based on side lengths.
- Add a method in the Rectangle class to determine if it's a square based on side lengths.
- Finally, create instances of both Rectangle and Triangle classes and calculate their areas.

Output:

```
Rectangle Area: 15.0
This Rectangle is not a Square.

Triangle Area: 45.5
This Triangle is a right-angled Triangle.
```

Task 4: Create a class hierarchy for employees in a company.

- Start with a base class called "Employee" with attributes "name", "employeeId", and "salary".
- Introduce a method in the Employee class to count the vowels in the employee’s name.
- Then, create two subclasses "Manager" and "Developer" that inherit from the Employee class.
- Add specific attributes and methods for each subclass.
- Finally, create instances of both Manager and Developer classes and display their information.

Output:

```
Manager Information:
Name: Sasuke and It contains 3 vowels.
Employee ID: 101
Salary: $80000.0
Department: HR

Developer Information:
Name: Naruto and It contains 3 vowels.
Employee ID: 201
Salary: $70000.0
Programming Language: Java
```

Task 5: Create a class hierarchy to model different types of electronic devices.

- Start with a base class called "Device" with attributes "brand" and "totalUnit", along with a method to display device information.
- Add a method in the Device class to compute the sum of all the digits in the totalUnit variable.
- Then, create two subclasses "Phone" and "Laptop" that inherit from the Device class.
- Add specific attributes and methods for each subclass.
- Finally, create instances of both Phone and Laptop classes and demonstrate the inheritance and method overriding.

Output:

```
Phone Information:
Brand: Apple
Total Unit: 150
Operating System: iOS
The sum of 150 is 6.

Laptop Information:
Brand: Dell
Total Unit: 125
Screen Size: 15.6 inches
The sum of 125 is 8.
```

Task 6: Create a class hierarchy to model different types of beverages.

- Start with a base class called "Beverage" with attributes "name" and "price", along with a method to display beverage information.
- Add a method in the Beverage class to find the duplicate char in the name attribute.
- Then, create two subclasses "Coffee" and "Soda" that inherit from the Beverage class.
- Add specific attributes and methods for each subclass.
- Finally, create instances of both Coffee and Soda classes and demonstrate the inheritance and method overriding.

Output:

```
Coffee Information:
Name: Espresso
Price: $3.5
Caffeine Level: 80 mg
Duplicate Character: [e, s]

Soda Information:
Name: Coca-Cola
Price: $1.99
Diet: No
Duplicate Character: [c, o, a]
```