EASTERN UNIVERSITY, SRI LANKA

FACULTY OF SCIENCE

First Year Second Semester Examination in Science - 2021/ 2022

(Aug./ Sep., 2024)

CS 1032 – Object Oriented Programming (Practical)

Answer All Questions **Time allowed: Two hours**

➢ Create a folder on your desktop with your index number (e.g., PS 0000) and save all your project files and outputs, including the question numbers, in that folder.

Q1.

Create a class named *burger* that stores information about a single burger. It should contain the following:

- Private instance variables to store the size of the burger (either small, medium, or large), the number of cheese fillings, the number of patty fillings, and the number of vegetable fillings.
- Constructor(s) that set all of the instance variables.
- Public methods to get and set the instance variables.
- A public method names TotCost() that returns a double that is the cost of the burger
- Burger cost is determined by:
- Large: Rs. 2145 + Rs. 245 per extra toppings
- Medium: Rs. 1845+ Rs. 215 per extra toppings
- Small: Rs. 1450 + Rs. 185 per extra toppings
- Public method named getDetail() that returns a string containing the burger size, quantity of each fillings.

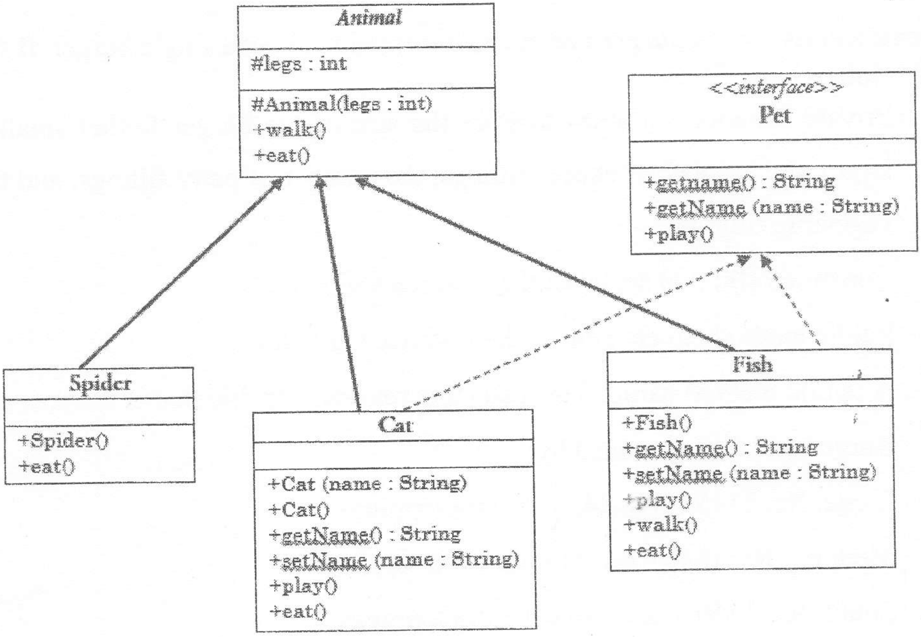
a) Write a test code to create several *burger* and output their descriptions.
(Hint: For an example, a large burger with ***two*** *cheese,* ***one*** *patty fillings should cost a total of RS. 2880.)*

b) Now create a *specialMeatBurger* class that allows up to three burgers to be saved in an order. Each burger saved should be a burger object.

c) Create a method TotCost() that returns the cost of order. In the runner order three burgers and return the total cost.

Q2.

Create a hierarchy of animals that is rooted in an abstract class *Animal.* Several of the animal classes will implement an interface called pet. It will be experimented with variations of these animals, their methods, and polymorphism.



a) Create the *Animal* class, which is the abstract superclass of all animals.

- Declare a protected integer attribute called legs, which records the number of legs for this animal.
- Define a protected constructor that initializes the legs attribute.
- Declare an abstract method eat.
- Declare a concrete method walk that prints out something about how the animal walks (include the number of legs).

b) Create the Spider class.

- The *Spider* class extends *Animal* class.

- Define a default constructor that calls the superclass constructor to specify that all spiders have eight legs.
- Implement the eat method.

c) Create the *pet* interface specified by UML diagram.

d) Create the *Cat* class that extends *Animal* and implements *Pet.*

- This class must include a string attribute to store the name of the pet.
- Define a constructor that takes one string parameter that specifies the name of the cat. This constructor must also call the super class constructor to specify that all cats have four legs.
- Define another constructor that takes no parameters. Have this constructor call the previous constructor and pass an empty string as the argument. (*Hint: use this keyword)*
- Implement the pet interface methods.
- Implement the eat method.

e) Create the *fish* class. Override the animal methods to specify that fish can't walk and don't have legs.

f) Create a *testAnimals* class. Have the main method create and manipulate instances of the class you created above.

[60%]