


National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	Software Construction & Development	Course Code:	CS-3001
	Degree Program:	BS(SE)	Semester:	Fall 2023
	Exam Duration:	60 Minutes	Total Marks:	45
	Paper Date:	2 – Oct - 2023	Weight:	15.00%
	Section:	ALL	Page(s):	6
	Exam Type:	Midterm-I		

Student Name: _____ **Roll No.** _____ **Section:** _____

Instruction/Notes: Attempt all questions. Do not use pencil or red ink. In case of confusion or ambiguity make a reasonable assumption. **Do not attach any extra sheet.** Use extra sheet for rough work only

Question 1 [CLO-1]

10+10=20 points

(a) Find the exception handling related errors in the following code considering best practices and suggest correction:

<pre> class InvalidAgeException extends Exception { public InvalidAgeException(String message) { super(message); System.out.println("User defined exception"); } } public class CustomExceptionDemo { public static void main(String[] args) { FileReader reader = new FileReader("file.txt"); char[] buffer = new char[1024]; reader.read(buffer); reader.close(); try { validateAge(15); } catch(Exception e) { System.out.println("Caught an Exception: " + e.getMessage()); } catch (InvalidAgeException e) { System.out.println("Caught an InvalidAgeException: " + e.getMessage()); } } public static void validateAge(int age) throws ArithmeticException { validateAge2(age); } public static void validateAge2(int age) throws InvalidAgeException { if (age < 18) { throw new Exception("Age must be 18 or older."); } System.out.println("Valid age: " + age); } } </pre>	<p>Errors/Correction:</p>
--	----------------------------------

(b) Given the code snippets below, complete the **main** to: (1) first get a list of all the discounted products in inventory and (2) print the list showing product id and discount rate. You cannot change any of the classes or the code already

provided but assume that setters and getters are already available. Also note that there is no specific order in which Discounted Products can be added to Inventory.

```
class Product {
    protected int id;
    protected String name;
    protected double price;
    public Product(int id, String name, double price) {
        // set appropriate data members ...
    }

    // setters and getters go here ...
}

class DiscountedProduct extends Product{
    protected float discount;

    public DiscountedProduct(int id, String name, double price, float discount) {
        super(id,name,price);
        this.discount = discount;
    }

    // setters and getters go here ...
}

class Inventory<T extends Product> {
    private Map<Integer, T> products = new HashMap<>();
    public void addProduct(T product) {
        products.put(product.getId(), product);
    }
    public List<T> getAllProducts() {
        return new ArrayList<T>(products.values());
    }
}

public class Main {
    public static void main(String[] args) {
        Inventory<Product> inventory = new Inventory<>();
        inventory.addProduct(new Product(1, "Laptop", 899.99));
        inventory.addProduct(new DiscountedProduct(2, "Headphones", 79.99,0.075));
        inventory.addProduct(new DiscountedProduct(3, "Charger", 99.99,0.05));
        inventory.addProduct(new Product(4, "Smartphone", 499.99));

        // other products and discounted products may be added here ...

        // complete code to get a list of all discounted products from inventory

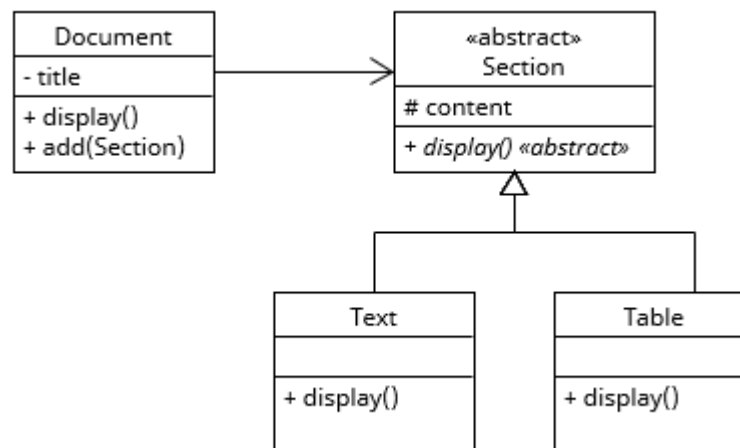
        // write code to show discount rate against each discounted product

    }
}
```

Question 2 [CLO-4]**5+5=10 points**

(a) A developer committed some unnecessary changes to an existing code file (Main.java) in the Git repository by mistake. (S)he wants to undo those changes and move the repository back to its previous state. What steps are required to accomplish this goal? Describe the strategy in general.

(b) What are the advantages of using Git?

Question 3 [CLO-1]**15 points**

Consider a simple document editing system using the concepts of object-orientation and a simple markup language that annotates the content. **Document** is organized in form of multiple **Section** subtypes. A **Text** is a section that displays the content in form of paragraphs. Content for **Text** may contain multiple paragraphs each separated using a simple markup **###**. Moreover a **Table** is a section that displays the content in form of rows and columns. Markup for tabular content can be [{ , , }, { , , }] where square brackets (i.e. []) contain rows and curly braces (i.e. {}) contains columns, both separated by comma (,).

Some examples are given below:

Content	Output
It is a light-weight markup language for creating documents.###It is inspired from many other markup languages used to produce diverse and professional documents.###However, it is just experimental.	It is a light-weight markup language for creating documents. It is inspired from many other markup languages used to produce diverse and professional documents. However, it is just experimental.
[{word,synonym},{document,record},{deed},{language,expression}]	word synonym document record deed language expression

All sections shall be displayed in the order they appear in the document.

Write Java code to implement the given UML diagram and make this Document Editor possible

