Rajalakshmi Engineering College

Name: Anu shalini B
Email: 240701043@rajalakshmi.edu.in
Roll no: 240701043
Phone: 9486176983
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 8_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 13

Section 1 : MCQ

1. What will be the output for the following code?

```
import java.io.*;

class OutOfStockException extends Exception {
    public OutOfStockException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            int stock = 0;
            if (stock == 0) {
                throw new OutOfStockException("Item is out of stock");
```

```
        }
    } catch (OutOfStockException e) {
        System.out.println(e.getMessage());
    }
  }
}
```

Answer

Item is out of stock

Status : Correct ***Marks : 1/1***

2. Which keyword is used to explicitly throw a custom exception?

Answer

throw

Status : Correct ***Marks : 1/1***

3. what is the output of the following code?

```
class MyException extends Exception {
  public MyException(String message) {
    super(message);
  }
}

class Test {
  static void check() throws MyException {
    throw new MyException("Custom Exception Occurred");
  }

  public static void main(String[] args) {
    try {
      check();
    } catch (Exception e) {
      System.out.println(e.getMessage());
    }
```

```
    }
}
```

Answer

Custom Exception Occurred

Status : Correct ***Marks : 1/1***

4. What will be the output for the following code?

```
class InvalidVotingAgeException extends Exception {
    public InvalidVotingAgeException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            int age = 15;
            if (age < 18) {
                throw new InvalidVotingAgeException("You are not eligible to
vote");
            }
            System.out.println("Eligible to vote");
        } catch (InvalidVotingAgeException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Answer

You are not eligible to vote

Status : Correct ***Marks : 1/1***

5. How do you create an unchecked custom exception?

Answer

By extending RuntimeException

Status : Correct **Marks : 1/1**

6. What will be the output of the following code?

```
class MyException extends Exception {
    public MyException() {
        super("Default Exception Message");
    }
}

class Test {
    public static void main(String[] args) {
        try {
            throw new MyException();
        } catch (MyException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Answer

No output

Status : Wrong **Marks : 0/1**

7. What will be the output for the following code?

```
import java.io.*;

class UnderageException extends Exception {
    public UnderageException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
```

```
        try {
            int age = 17;
            if (age < 18) {
                throw new UnderageException("Underage, cannot proceed");
            }
        } catch (UnderageException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Answer

Underage, cannot proceed

Status : Correct ***Marks : 1/1***

8. Which of the following is true about custom exceptions?

Answer

Custom exceptions must extend either Exception or RuntimeException

Status : Correct ***Marks : 1/1***

9. What will be the output for the following code?

```
import java.io.*;

class NegativeAgeException extends Exception {
    public NegativeAgeException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            int age = -5;
            if (age < 0) {
```

```
            throw new NegativeAgeException("Age cannot be negative");
        }
    } catch (NegativeAgeException e) {
        System.out.println(e.getMessage());
    }
  }
}
```

Answer

Age cannot be negative

Status : Correct ***Marks : 1/1***

10. What will be the output for the following code?

```
import java.io.*;

class TemperatureTooHighException extends Exception {
  public TemperatureTooHighException(String message) {
    super(message);
  }
}

class Test {
  public static void main(String[] args) {
    try {
      int temperature = 110;
      if (temperature > 100) {
        throw new TemperatureTooHighException("Temperature too
high");
      }
    } catch (TemperatureTooHighException e) {
      System.out.println(e.getMessage());
    }
  }
}
```

Answer

Temperature too high

Status : Correct **Marks : 1/1**

11. What is the purpose of a custom exception in Java?

Answer

To create user-defined exceptions for specific scenarios

Status : Correct **Marks : 1/1**

12. What will be the output for the following code?

```
class InvalidUsernameException extends Exception {
    public InvalidUsernameException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            String username = "abc";
            if (username.length() < 5) {
                throw new InvalidUsernameException("Username must be at
least 5 characters long");
            }
        } catch (InvalidUsernameException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

Answer

Username must be at least 5 characters long

Status : Correct **Marks : 1/1**

13. what is the output of the following code?

```
class MyException extends Exception {
    public MyException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            throw new MyException("Error occurred");
        } catch (MyException e) {
            System.out.println(e);
        }
    }
}
```

Answer

MyException: Error occurred

Status : Correct ***Marks : 1/1***

14. What will happen if a checked custom exception is thrown inside a method without being caught or declared?

Answer

Compilation Error

Status : Correct ***Marks : 1/1***

15. What will be the output for the following code?

```
class NegativeBalanceException extends Exception {
    public NegativeBalanceException(String message) {
        super(message);
    }
}

class Test {
```

```
    public static void main(String[] args) {
        try {
            double balance = -500;
            if (balance < 0) {
                throw new NegativeBalanceException("Balance cannot be
negative");
            }
        } catch (NegativeBalanceException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Answer

null

Status : Wrong ***Marks : 0/1***

Rajalakshmi Engineering College

Name: Anu shalini B
Email: 240701043@rajalakshmi.edu.in
Roll no: 240701043
Phone: 9486176983
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

A local municipality is implementing an online voting system for a community event and wants to ensure that only eligible voters (those aged 18 or older) can participate.

Your task is to develop a program that validates the age of individuals attempting to vote online. If the user's age is below 18, the program should throw a custom exception, InvalidAgeException, preventing them from casting their vote. If the input is invalid, catch the appropriate InputMismatchException and print the in-built exception message.

Input Format

The input consists of an integer representing the age.

Output Format

If the age is 18 or older, print "Eligible to vote"

If the age is below 18, print "Exception occurred: InvalidAgeException: Age is not valid to vote"

If there is any other type of exception, print "An error occurred: " followed by the in-built exception message.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 20
Output: Eligible to vote

Answer

```
import java.util.InputMismatchException;
import java.util.Scanner;
class InvalidAgeException extends Exception {
    public InvalidAgeException(String message) {
        super(message);
    }
}
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            int age = sc.nextInt();

            if (age < 18) {
                throw new InvalidAgeException("Age is not valid to vote");
            } else {
                System.out.println("Eligible to vote");
            }

        } catch (InvalidAgeException e) {
            System.out.println("Exception occurred: InvalidAgeException: " +
e.getMessage());
        } catch (InputMismatchException e) {
            System.out.println("An error occurred: " + e.getClass().getName());
```

```
        } catch (Exception e) {
            System.out.println("An error occurred: " + e.getMessage());
        } finally {
            sc.close();
        }
    }
}
```

Status : Correct **Marks : 10/10**

Rajalakshmi Engineering College

Name: Anu shalini B
Email: 240701043@rajalakshmi.edu.in
Roll no: 240701043
Phone: 9486176983
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 8_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a user registration system, there is a requirement to implement a username validation module. Users attempting to register must adhere to specific criteria for their usernames to be considered valid.

Your task is to develop a program that takes user input for a desired username and validates it according to the following rules:

The username must not contain any spaces.The username must be at least 5 characters long.

Implement a custom exception, InvalidUsernameException, to handle cases where the entered username does not meet the specified criteria.

Input Format

The input consists of a string S, representing the desired username.

Output Format

If the username is valid, print "Username is valid: [S]".

If the username is invalid:

1. If the username is short, print "Invalid Username: Username must be at least 5 characters long"
2. If the username contains spaces, print "Invalid Username: Username cannot contain spaces"

Refer to the sample output for formatting specifications.

Sample Test Case

Input: John

Output: Invalid Username: Username must be at least 5 characters long

Answer

```
import java.util.Scanner;
class InvalidUsernameException extends Exception {
    public InvalidUsernameException(String message) {
        super(message);
    }
}

public class Main {
    public static void validateUsername(String username) throws
InvalidUsernameException {
        if (username.contains(" ")) {
            throw new InvalidUsernameException("Invalid Username: Username
cannot contain spaces");
        } else if (username.length() < 5) {
            throw new InvalidUsernameException("Invalid Username: Username must
be at least 5 characters long");
        } else {
            System.out.println("Username is valid: " + username);
        }
    }
```

```
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String username = sc.nextLine().trim();

        try {
            validateUsername(username);
        } catch (InvalidUsernameException e) {
            System.out.println(e.getMessage());
        }

        sc.close();
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Anu shalini B
Email: 240701043@rajalakshmi.edu.in
Roll no: 240701043
Phone: 9486176983
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

Maria, a software developer, is working on an inventory management system project using Java that utilizes an inventory interface to manage a store's products.

The interface should define two methods: addProduct, which adds a product by accepting its name, price, and quantity, and calculateTotalValue, which computes the total value of all products in the inventory. Implement the interface in a class called SimpleInventory, which internally manages a list of Product objects.

Each Product object should encapsulate the product's name, price, and quantity and include a method to calculate its value as price × quantity. The system should allow users to dynamically add products to the inventory and calculate the total value of all products stored.

Help Maria achieve the task.

Input Format

The first line of input consists of an integer to choose one of the following options:

1 - to add a product to the inventory.

2 - to calculate and view the total inventory value.

3 - to exit the program.

For Choice 1 (Add Product):

The next input line is the string representing the product name as a string (single or multi-word, without quotes).

The next line is a double value representing the price as a decimal value

The next line is an integer value representing the quantity as an integer

For Choices 2 and 3, no additional input is required

Output Format

The output displays the results of the commands as follows:

- For the addProduct command, the program should display "Product added to inventory."
- For choice 2, the program should display "Total inventory value [totalvalue]. "The total value should be displayed with one decimal place. If there is no product in the inventory, print the total as 0.0.
- For choice 3, the program should exit

If the choice is not 1, 2, or 3, then print "Invalid choice. Please select a valid option (1/2/3).".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1
Laptop
800.0
3
2
5
3

Output: Product added to inventory.
Total inventory value: $2400.0
Invalid choice. Please select a valid option (1/2/3).

Answer

```
import java.util.Scanner;
import java.util.*;

interface Inventory {
    void addProduct(String name, double price, int quantity);
    double calculateTotalValue();
}

class Product {
    private String name;
    private double price;
    private int quantity;

    public Product(String name, double price, int quantity) {
        this.name = name;
        this.price = price;
        this.quantity = quantity;
    }

    public double getValue() {
        return price * quantity;
    }
}

class SimpleInventory implements Inventory {
    private List<Product> products = new ArrayList<>();

    public void addProduct(String name, double price, int quantity) {
```

```
        products.add(new Product(name, price, quantity));
        System.out.println("Product added to inventory.");
    }

    public double calculateTotalValue() {
        double total = 0.0;
        for (Product p : products) {
            total += p.getValue();
        }
        return total;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Inventory inventory = new SimpleInventory(10);
        while (true) {
            int choice = scanner.nextInt();
            if (choice == 1) {
                scanner.nextLine();
                String productName = scanner.nextLine();
                double price = scanner.nextDouble();
                int quantity = scanner.nextInt();
                inventory.addProduct(productName, price, quantity);
            } else if (choice == 2) {
                double totalValue = inventory.calculateTotalValue();
                System.out.println("Total inventory value: $" + totalValue);
            } else if (choice == 3) {
                break;
            } else {
                System.out.println("Invalid choice. Please select a valid option
(1/2/3).");
            }
        }
        scanner.close();
    }
}
```

Status : Wrong **Marks : 0/10**

Rajalakshmi Engineering College

Name: Anu shalini B
Email: 240701043@rajalakshmi.edu.in
Roll no: 240701043
Phone: 9486176983
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

A financial analyst, Alex, needs a program to calculate simple interest for various financial transactions. He requires a straightforward tool that takes in the principal amount, interest rate, and time in years and computes the interest.

The formula to be used is: Interest = Principal × Rate × Time / 100

Implement this functionality using the InterestCalculator interface and the SimpleInterestCalculator class.

Input Format

The first line of input consists of the principal amount P as a double value.

The second line of input consists of the annual interest rate r as a double value.

The third line of input consists of the number of years t as a positive integer, which is an integer value.

Output Format

The output displays the calculated simple interest in the following format: "Simple Interest: [interest_value]", Here, [interest_value] should be replaced with the actual interest value calculated by the program.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1000.00
5.00
2
Output: Simple Interest: 100.0

Answer

```
import java.util.Scanner;


interface InterestCalculator {
    double calculateInterest(double principal, double rate, int time);
}

class SimpleInterestCalculator implements InterestCalculator {
    public double calculateInterest(double principal, double rate, int time) {
        return (principal * rate * time) / 100;
    }
}



class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        double principal = scanner.nextDouble();
```

```
        double rate = scanner.nextDouble();
        int time = scanner.nextInt();
        InterestCalculator calculator = new SimpleInterestCalculator();
        double interest = calculator.simpleInterest(principal, rate, time);
        System.out.println("Simple Interest: " + interest);
    }
}
```

Status : Wrong **Marks : 0/10**

Rajalakshmi Engineering College

Name: Anu shalini B
Email: 240701043@rajalakshmi.edu.in
Roll no: 240701043
Phone: 9486176983
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

Jaheer is working on a health monitoring system to help individuals calculate their Body Mass Index (BMI). He has implemented a basic BMI calculator and an interface called HealthCalculator. It should have a method called calculateBMI.

You are tasked with creating a program that takes weight and height as input, calculates the BMI using the BMICalculator class, and displays the result. If the height or weight is less than or equal to zero, then return -1.

Formula: BMI = weight / (height * height)

Input Format

The first line of input consists of a double value W, the person's weight in kilograms.

The second line consists of a double value H, the height of the person in meters.

Output Format

The output displays "BMI: " followed by a double value, representing the calculated BMI, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 70.0
1.75
Output: BMI: 22.86

Answer

```
import java.util.Scanner;

interface HealthCalculator{
   double calculateBMI(double weight,double height);
}
class BMIcalculator implements HealthCalculator{
   public double BMIcalculate (double weight,double height){
      if (weight <=0 || height <=0)return -1;
      return weight/(height * height);
   }
}


class Main {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      double weight = scanner.nextDouble();
      double height = scanner.nextDouble();

      BMICalculator bmiCalculator = new BMICalculator();

      double bmi = bmiCalculator.calculateBMI(weight, height);

      System.out.printf("BMI: %.2f\n", bmi);
```

```
        scanner.close();
    }
}
```

Status : Wrong

Marks : 0/10