BITA/7/23/049/TZ

QUESTION ONE

(Rectangle Class) Create a class Rectangle with attributes length and width. Provide member functions that calculate the perimeter and the area of the rectangle

PROGRAM AND OUTPUT

C: > Users > CHARMING > Desktop > kaz > Rectangle.java

```java
class Rectangle {
    private double length;
    private double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    public double calculatePerimeter() {
        return 2 * (length + width);
    }

    public double calculateArea() {
        return length * width;
    }

    public static void main(String[] args) {
        Rectangle rect = new Rectangle(5.0, 3.0);
        System.out.println("Perimeter: " + rect.calculatePerimeter());
        System.out.println("Area: " + rect.calculateArea());
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\CHARMING> cd desktop
PS C:\Users\CHARMING\desktop> cd kaz
PS C:\Users\CHARMING\desktop\kaz> javac Rectangle.java
PS C:\Users\CHARMING\desktop\kaz> java Rectangle
Perimeter: 16.0
Area: 15.0
PS C:\Users\CHARMING\desktop\kaz>
```

QUESTION TWO

(Invoice Class) Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as data members a part number (type string), a part description (type string), a quantity of the item being purchased (type int) and a price per item (type int). Your class should have a constructor that initializes the four data members. In addition, provide a member function named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as an int value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0. Write a test program that demonstrates class Invoice's capabilities.

PROGRAM AND OUTPUT

```java
class Invoice {
    private String partNumber;
    private String partDescription;
    private int quantity;
    private int pricePerItem;

    public Invoice(String partNumber, String partDescription, int quantity, int pricePerItem) {
        this.partNumber = partNumber;
        this.partDescription = partDescription;
        this.quantity = Math.max(quantity, 0);
        this.pricePerItem = Math.max(pricePerItem, 0);
    }

    public int getInvoiceAmount() {
        return quantity * pricePerItem;
    }

    public static void main(String[] args) {
        Invoice invoice = new Invoice("1234", "Hammer", 5, 20);
        System.out.println("Invoice Amount: $" + invoice.getInvoiceAmount());
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\CHARMING> cd desktop
PS C:\Users\CHARMING\desktop> cd kaz
PS C:\Users\CHARMING\desktop\kaz> javac Invoice.java
PS C:\Users\CHARMING\desktop\kaz> java Invoice
Invoice Amount: $100
PS C:\Users\CHARMING\desktop\kaz>
```

QUESTION THREE

(Account Class) Create a class called Account that a bank might use to represent customers' bank accounts. Your class should include one data member of type int to represent the account balance. Your class should provide a constructor that receives an initial balance and uses it to initialize the data member. The constructor should validate the initial balance to ensure that it is greater than or equal to 0. If not, the balance should be set to 0 and the constructor should display an error message, indicating that the initial balance was invalid. The class should provide three member functions. Member function credit should add an amount to the current balance. Member function debit should withdraw money from the Account and should ensure that the debit amount does not exceed the Account's balance. If it does, the balance should be left unchanged and the function should print a message indicating "Debit amount exceeded account balance." Member function getBalance should return the current balance. Create a program that creates two Account objects and tests the member functions of class Account.

PROGRAM

```java
class Account {
    private int balance;

    public Account(int initialBalance) {
        if (initialBalance >= 0) {
            this.balance = initialBalance;
        } else {
            this.balance = 0;
            System.out.println("Error: Initial balance invalid.");
        }
    }

    public void credit(int amount) {
        balance += amount;
    }

    public void debit(int amount) {
        if (amount > balance) {
            System.out.println("Debit amount exceeded account balance.");
        } else {
            balance -= amount;
        }
    }

    public int getBalance() {
        return balance;
    }
```

OPROGRAM AND OUTPUT

```java
    public static void main(String[] args) {
        Account account1 = new Account(100);
        Account account2 = new Account(-50);

        account1.credit(50);
        account1.debit(30);
        System.out.println("Account1 Balance: $" + account1.getBalance());

        account2.credit(100);
        account2.debit(200);
        System.out.println("Account2 Balance: $" + account2.getBalance());
    }
}
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\CHARMING\desktop> cd kaz
PS C:\Users\CHARMING\desktop\kaz> javac Account.java
PS C:\Users\CHARMING\desktop\kaz> java Account
Error: Initial balance invalid.
Account1 Balance: $120
Debit amount exceeded account balance.
Account2 Balance: $100
PS C:\Users\CHARMING\desktop\kaz>
```

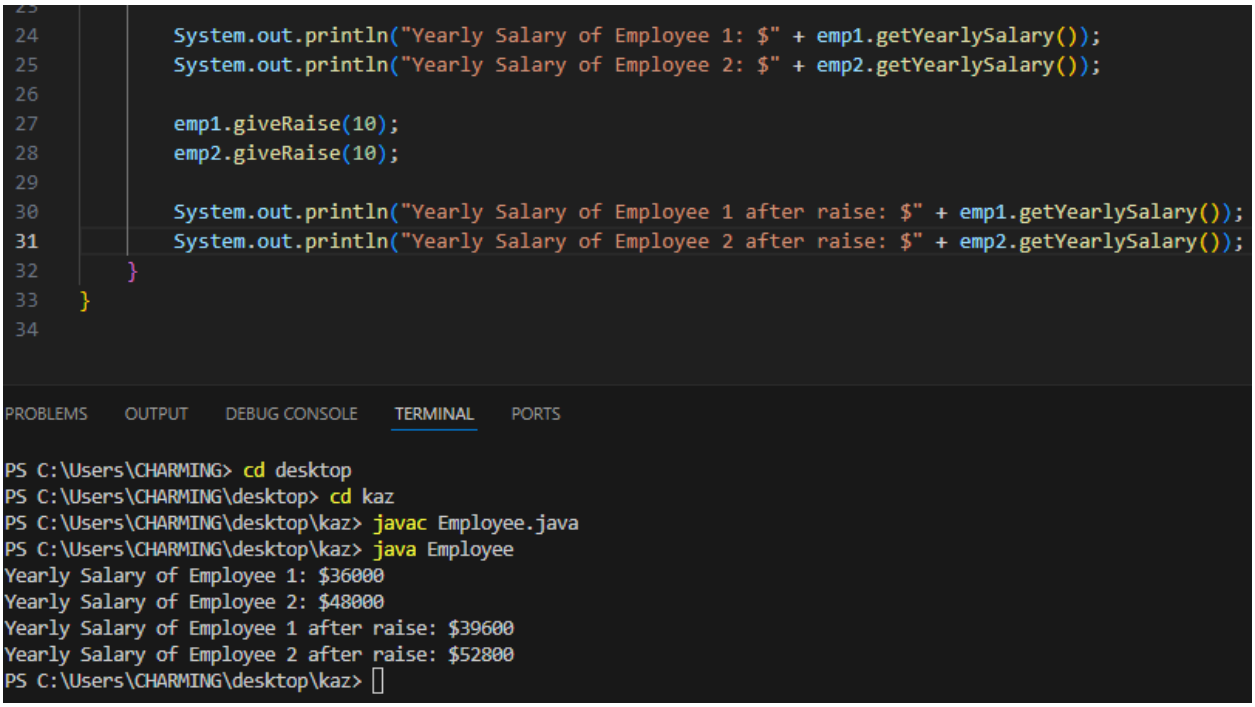
QUESTION FOUR

(Employee Class) Create a class called Employee that includes three pieces of information as data members a first name (type string), a last name (type string) and a monthly salary (type int). Your class should have a constructor that initializes the three data members. If the monthly salary is not positive, set it to 0. Write a test program that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10 percent raise and display each Employee's yearly salary again.

PROGRAM

```java
class Employee {
    private String firstName;
    private String lastName;
    private int monthlySalary;

    public Employee(String firstName, String lastName, int monthlySalary) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.monthlySalary = Math.max(monthlySalary, 0);
    }

    public int getYearlySalary() {
        return monthlySalary * 12;
    }

    public void giveRaise(double percentage) {
        monthlySalary += (int)(monthlySalary * percentage / 100);
    }

    public static void main(String[] args) {
        Employee emp1 = new Employee("John", "Doe", 3000);
        Employee emp2 = new Employee("Jane", "Smith", 4000);
```

PROGRAM AND OUTPUT



```
        System.out.println("Yearly Salary of Employee 1: $" + emp1.getYearlySalary());
        System.out.println("Yearly Salary of Employee 2: $" + emp2.getYearlySalary());

        emp1.giveRaise(10);
        emp2.giveRaise(10);

        System.out.println("Yearly Salary of Employee 1 after raise: $" + emp1.getYearlySalary());
        System.out.println("Yearly Salary of Employee 2 after raise: $" + emp2.getYearlySalary());
    }
}

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\CHARMING> cd desktop
PS C:\Users\CHARMING\desktop> cd kaz
PS C:\Users\CHARMING\desktop\kaz> javac Employee.java
PS C:\Users\CHARMING\desktop\kaz> java Employee
Yearly Salary of Employee 1: $36000
Yearly Salary of Employee 2: $48000
Yearly Salary of Employee 1 after raise: $39600
Yearly Salary of Employee 2 after raise: $52800
PS C:\Users\CHARMING\desktop\kaz>
```

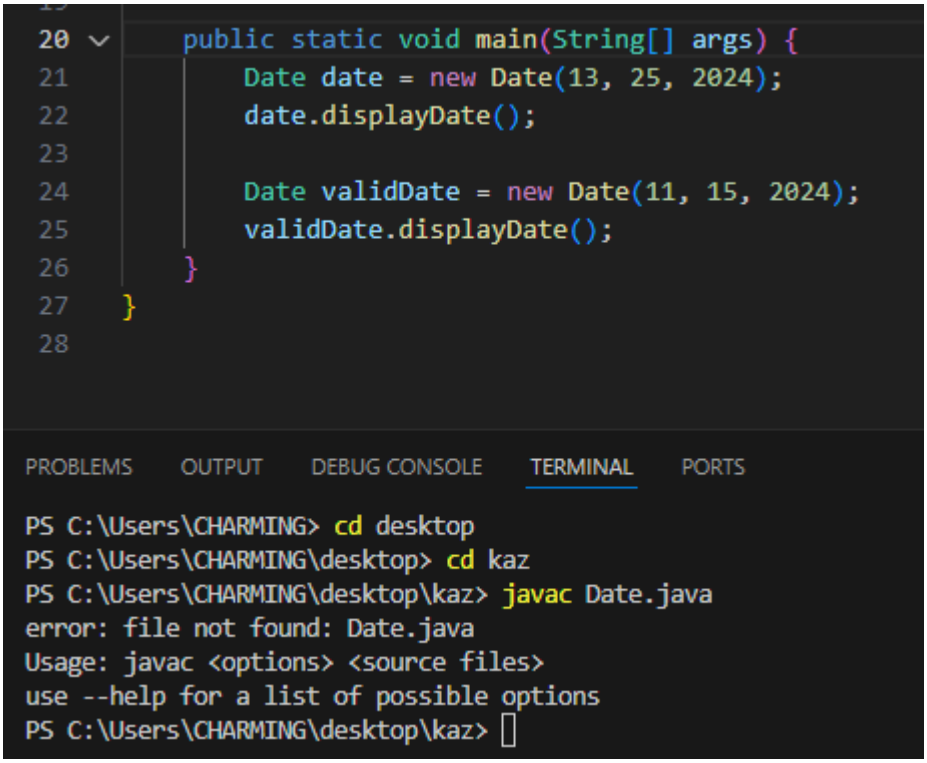
QUESTION FIVE

(Date Class) Create a class called Date that includes three pieces of information as data members a month (type int), a day (type int) and a year (type int). Your class should have a constructor with three parameters that uses the parameters to initialize the three data members. For the purpose of this exercise, assume that the values provided for the year and day are correct, but ensure that the month value is in the range 1-12; if it is not, set the month to 1. Provide a member function displayDate that displays the month, day and year separated by forward slashes (/). Write a test program that demonstrates class Date's capabilities

PROGRAM

```java
class Date {
    private int month;
    private int day;
    private int year;

    public Date(int month, int day, int year) {
        if (month >= 1 && month <= 12) {
            this.month = month;
        } else {
            this.month = 1;
        }
        this.day = day;
        this.year = year;
    }

    public void displayDate() {
        System.out.println(month + "/" + day + "/" + year);
    }

    public static void main(String[] args) {
        Date date = new Date(13, 25, 2024);
        date.displayDate();
```

PROGRAM AND OUTPUT



```
    public static void main(String[] args) {
        Date date = new Date(13, 25, 2024);
        date.displayDate();

        Date validDate = new Date(11, 15, 2024);
        validDate.displayDate();
    }
}

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\CHARMING> cd desktop
PS C:\Users\CHARMING\desktop> cd kaz
PS C:\Users\CHARMING\desktop\kaz> javac Date.java
error: file not found: Date.java
Usage: javac <options> <source files>
use --help for a list of possible options
PS C:\Users\CHARMING\desktop\kaz>
```