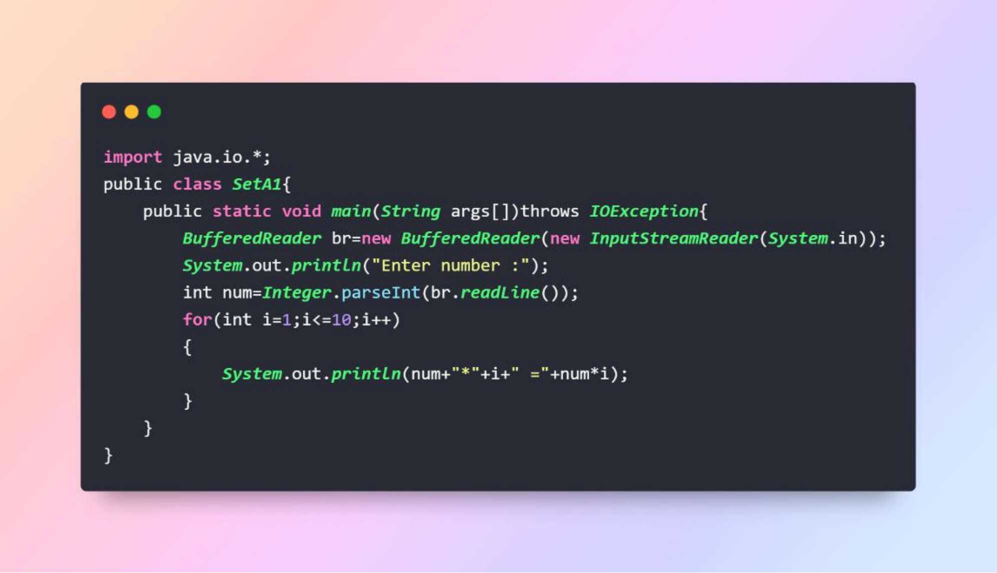
Java Programming

Assignment 1

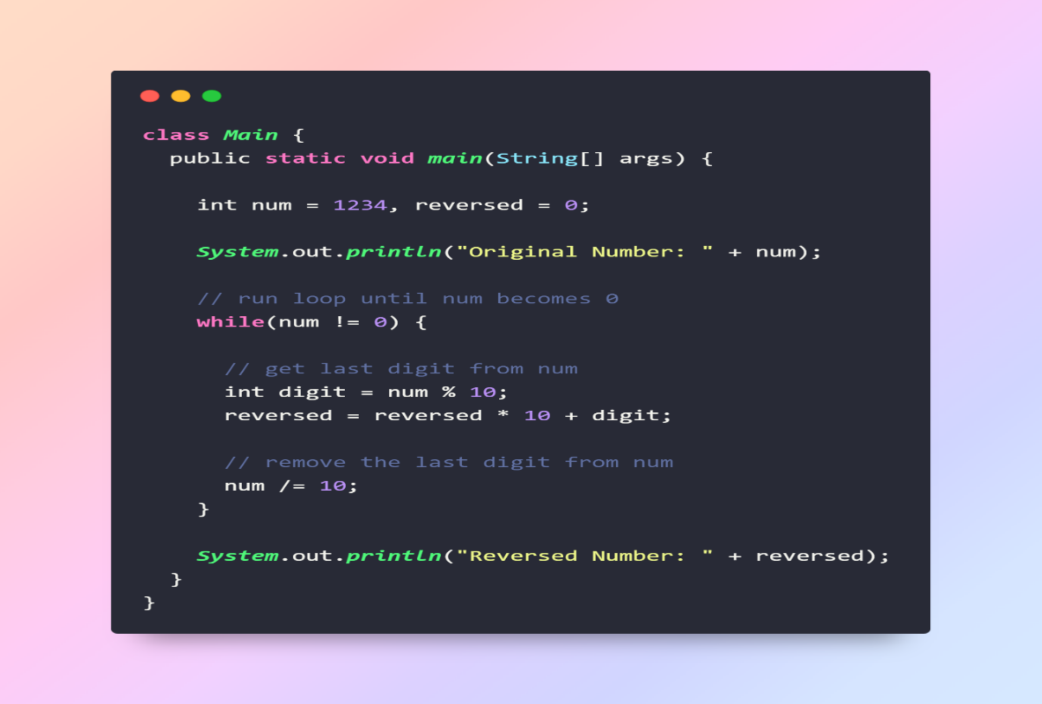
SET A

1. Write a Java program to accept a number from command prompt and generate multiplication table of a number. Accept number using BufferedReader class.



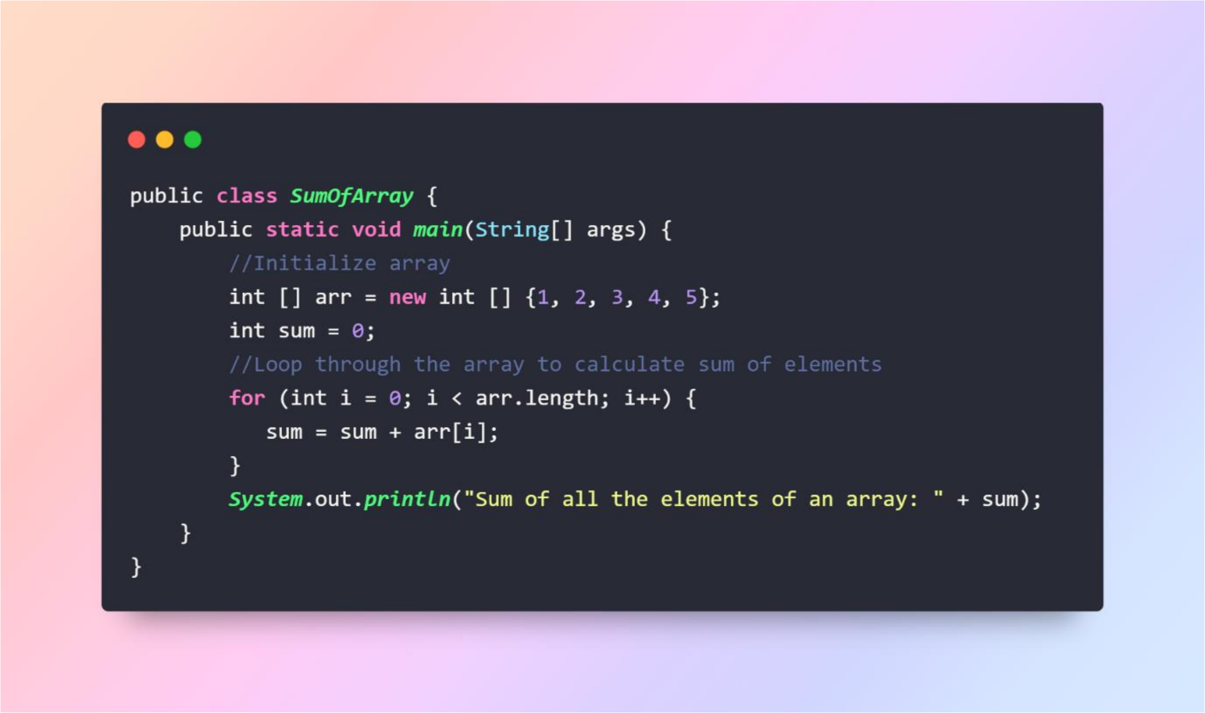
```
import java.io.*;
public class SetA1{
    public static void main(String args[])throws IOException{
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Enter number :");
        int num=Integer.parseInt(br.readLine());
        for(int i=1;i<=10;i++)
        {
            System.out.println(num+"*"+i+" ="+num*i);
        }
    }
}
```

2. Write a Java Program to Reverse a Number. Accept number using command line argument.



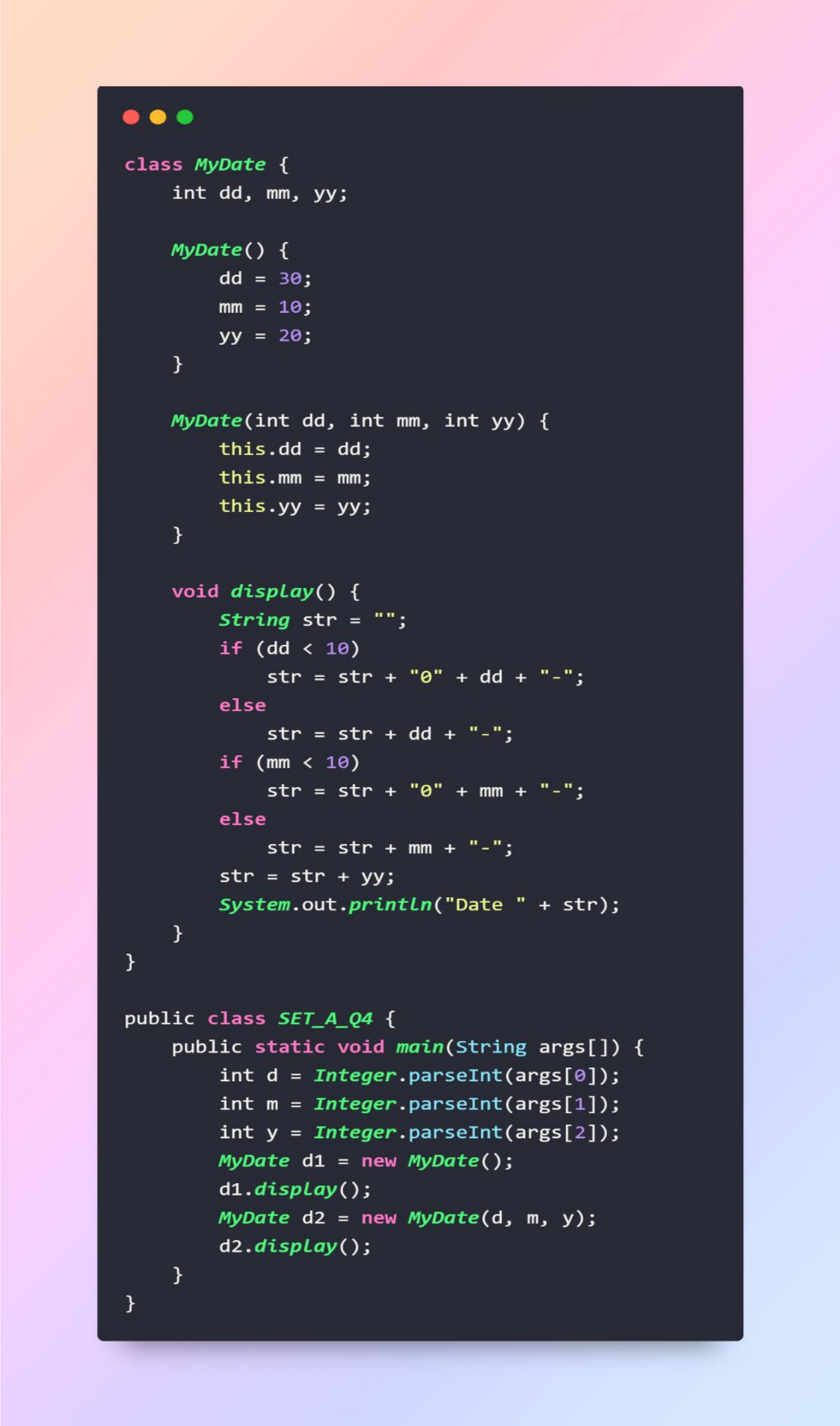
```
class Main {
  public static void main(String[] args) {

    int num = 1234, reversed = 0;

    System.out.println("Original Number: " + num);

    // run loop until num becomes 0
    while(num != 0) {

      // get last digit from num
      int digit = num % 10;
      reversed = reversed * 10 + digit;

      // remove the last digit from num
      num /= 10;
    }

    System.out.println("Reversed Number: " + reversed);
  }
}
```

3. Write a Java program to print the sum of elements of the array. Also display array elements in ascending order.



```
public class SumOfArray {
    public static void main(String[] args) {
        //Initialize array
        int [] arr = new int [] {1, 2, 3, 4, 5};
        int sum = 0;
        //Loop through the array to calculate sum of elements
        for (int i = 0; i < arr.length; i++) {
           sum = sum + arr[i];
        }
        System.out.println("Sum of all the elements of an array: " + sum);
    }
}
```

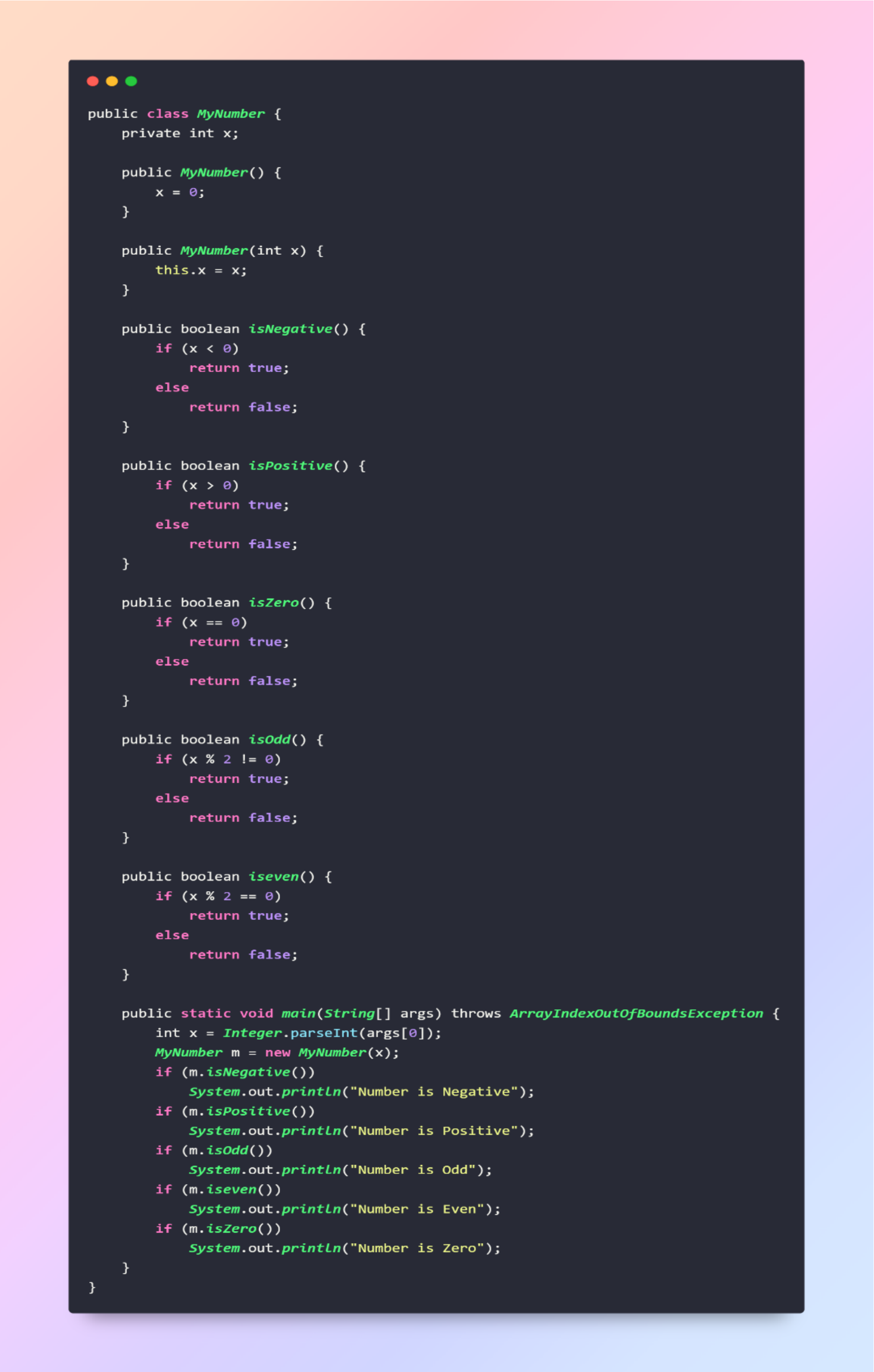
4. Write a Java program create class as MyDate with dd,mm,yy as data members. Write default and parameterized constructor. Display the date in dd-mm-yy format.(Use this keyword)



```java
class MyDate {
    int dd, mm, yy;

    MyDate() {
        dd = 30;
        mm = 10;
        yy = 20;
    }

    MyDate(int dd, int mm, int yy) {
        this.dd = dd;
        this.mm = mm;
        this.yy = yy;
    }

    void display() {
        String str = "";
        if (dd < 10)
            str = str + "0" + dd + "-";
        else
            str = str + dd + "-";
        if (mm < 10)
            str = str + "0" + mm + "-";
        else
            str = str + mm + "-";
        str = str + yy;
        System.out.println("Date " + str);
    }
}

public class SET_A_Q4 {
    public static void main(String args[]) {
        int d = Integer.parseInt(args[0]);
        int m = Integer.parseInt(args[1]);
        int y = Integer.parseInt(args[2]);
        MyDate d1 = new MyDate();
        d1.display();
        MyDate d2 = new MyDate(d, m, y);
        d2.display();
    }
}
```

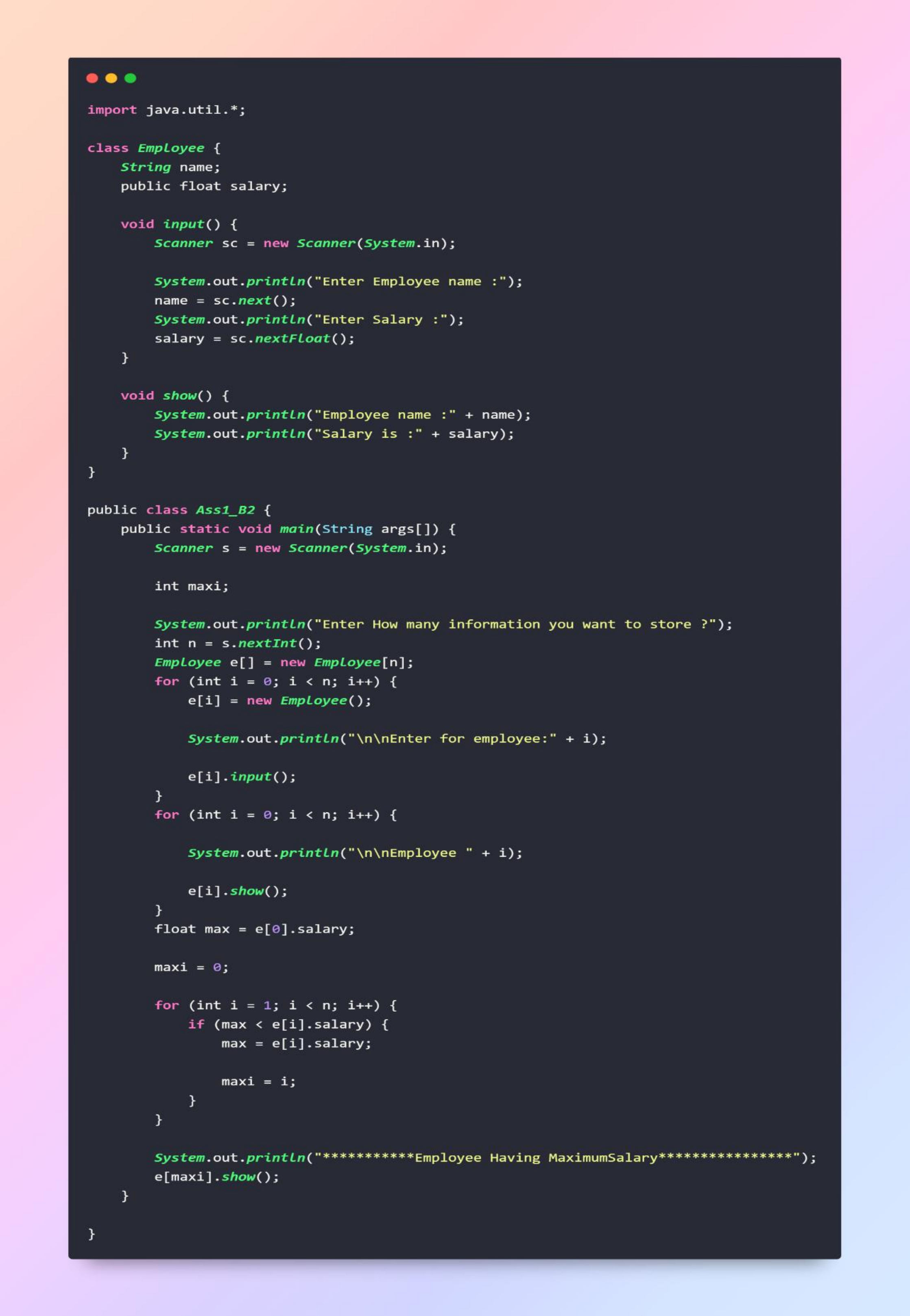
SET B

1. Define a class MyNumber having one private integer data member. Write a default constructor initialize it to 0 and another constructor to initialize it to a value. Write methods isNegative, isPositive, isOdd, iseven. Use command line argument to pass a value to the object and perform the above tests.



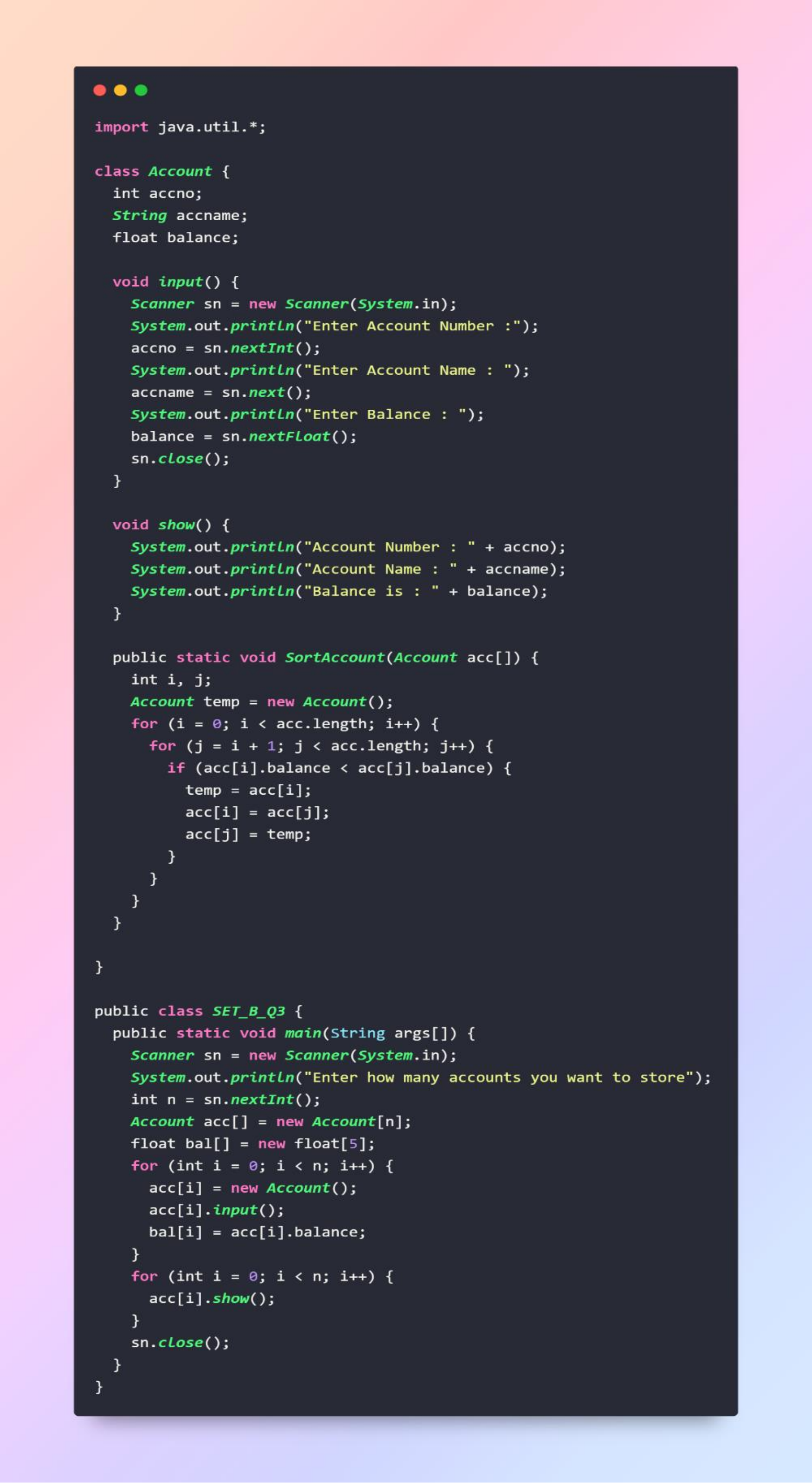
```
public class MyNumber {
    private int x;

    public MyNumber() {
        x = 0;
    }

    public MyNumber(int x) {
        this.x = x;
    }

    public boolean isNegative() {
        if (x < 0)
            return true;
        else
            return false;
    }

    public boolean isPositive() {
        if (x > 0)
            return true;
        else
            return false;
    }

    public boolean isZero() {
        if (x == 0)
            return true;
        else
            return false;
    }

    public boolean isOdd() {
        if (x % 2 != 0)
            return true;
        else
            return false;
    }

    public boolean iseven() {
        if (x % 2 == 0)
            return true;
        else
            return false;
    }

    public static void main(String[] args) throws ArrayIndexOutOfBoundsException {
        int x = Integer.parseInt(args[0]);
        MyNumber m = new MyNumber(x);
        if (m.isNegative())
            System.out.println("Number is Negative");
        if (m.isPositive())
            System.out.println("Number is Positive");
        if (m.isOdd())
            System.out.println("Number is Odd");
        if (m.iseven())
            System.out.println("Number is Even");
        if (m.isZero())
            System.out.println("Number is Zero");
    }
}
```

2. Write a java program which define class Employee with data member as name and salary. Program store the information of 5 Employees. (Use array of object)



```java
import java.util.*;

class Employee {
    String name;
    public float salary;

    void input() {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Employee name :");
        name = sc.next();
        System.out.println("Enter Salary :");
        salary = sc.nextFloat();
    }

    void show() {
        System.out.println("Employee name :" + name);
        System.out.println("Salary is :" + salary);
    }
}

public class Ass1_B2 {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);

        int maxi;

        System.out.println("Enter How many information you want to store ?");
        int n = s.nextInt();
        Employee e[] = new Employee[n];
        for (int i = 0; i < n; i++) {
            e[i] = new Employee();

            System.out.println("\n\nEnter for employee:" + i);

            e[i].input();
        }
        for (int i = 0; i < n; i++) {

            System.out.println("\n\nEmployee " + i);

            e[i].show();
        }
        float max = e[0].salary;

        maxi = 0;

        for (int i = 1; i < n; i++) {
            if (max < e[i].salary) {
                max = e[i].salary;

                maxi = i;
            }
        }

        System.out.println("***********Employee Having MaximumSalary****************");
        e[maxi].show();
    }

}
```

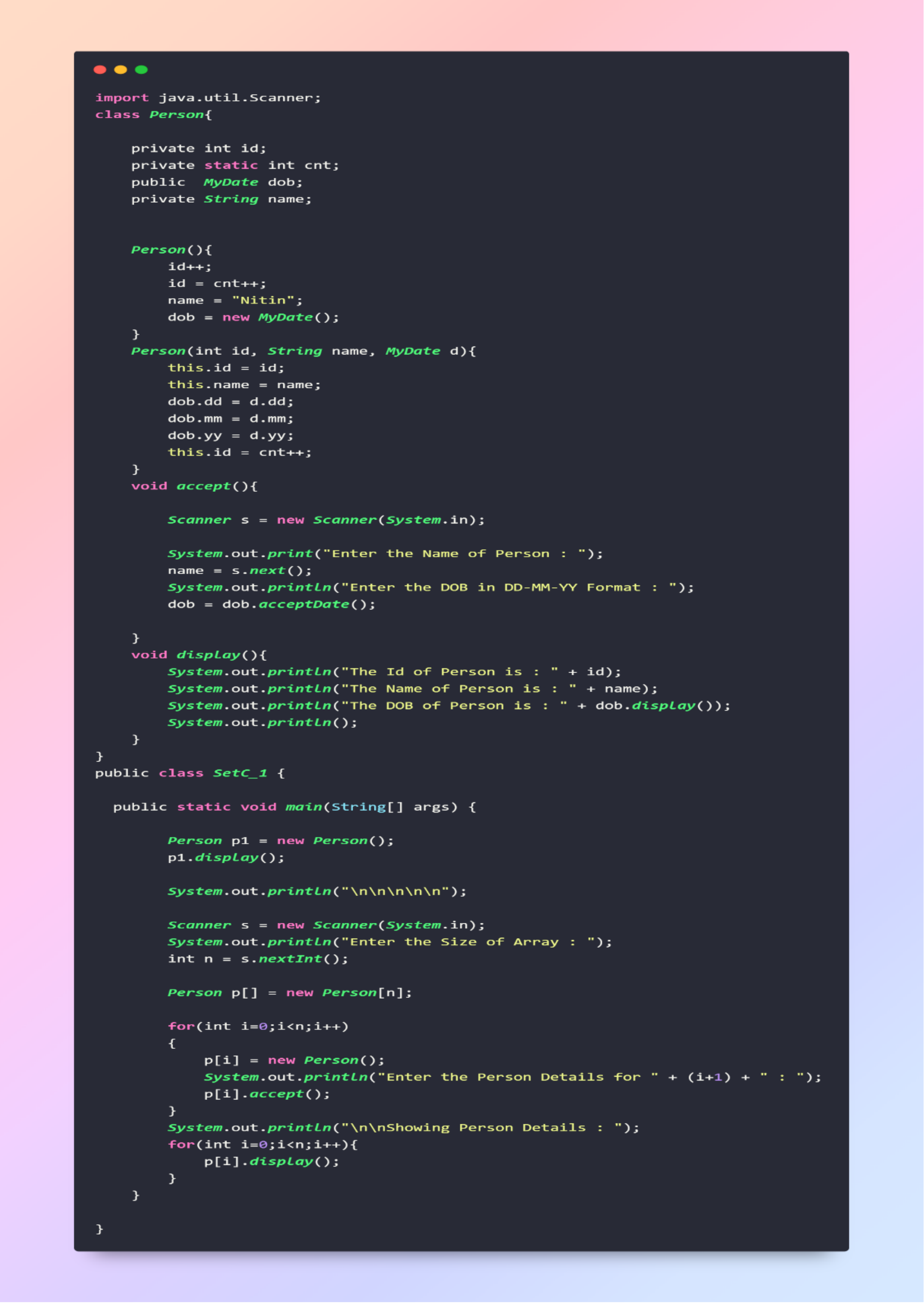
3. Write a java program to create class Account (accno, accname, balance). Create an array of “n” Account objects. Define static method “sortAccount” which sorts the array on the basis of balance. Display account details in sorted order.



```
import java.util.*;

class Account {
  int accno;
  String accname;
  float balance;

  void input() {
    Scanner sn = new Scanner(System.in);
    System.out.println("Enter Account Number :");
    accno = sn.nextInt();
    System.out.println("Enter Account Name : ");
    accname = sn.next();
    System.out.println("Enter Balance : ");
    balance = sn.nextFloat();
    sn.close();
  }

  void show() {
    System.out.println("Account Number : " + accno);
    System.out.println("Account Name : " + accname);
    System.out.println("Balance is : " + balance);
  }

  public static void SortAccount(Account acc[]) {
    int i, j;
    Account temp = new Account();
    for (i = 0; i < acc.length; i++) {
      for (j = i + 1; j < acc.length; j++) {
        if (acc[i].balance < acc[j].balance) {
          temp = acc[i];
          acc[i] = acc[j];
          acc[j] = temp;
        }
      }
    }
  }

}

public class SET_B_Q3 {
  public static void main(String args[]) {
    Scanner sn = new Scanner(System.in);
    System.out.println("Enter how many accounts you want to store");
    int n = sn.nextInt();
    Account acc[] = new Account[n];
    float bal[] = new float[5];
    for (int i = 0; i < n; i++) {
      acc[i] = new Account();
      acc[i].input();
      bal[i] = acc[i].balance;
    }
    for (int i = 0; i < n; i++) {
      acc[i].show();
    }
    sn.close();
  }
}
```

SET C

1. Define a class Person (id, name, dateofbirth). For dateofbirth create an object of MyDate class which is created in SET A 4. Define default and parameterized constructor. Also define accept and display function in Person and MyDate class. Call constructor and function of MyDate class from Person class for dateofbirth. Generate id automatically using static variable. Accept the details of n person and display the details. Sample code: public class Person { private int id; private String name; private MyDate dob; private static int cnt=1; ------- ------- }

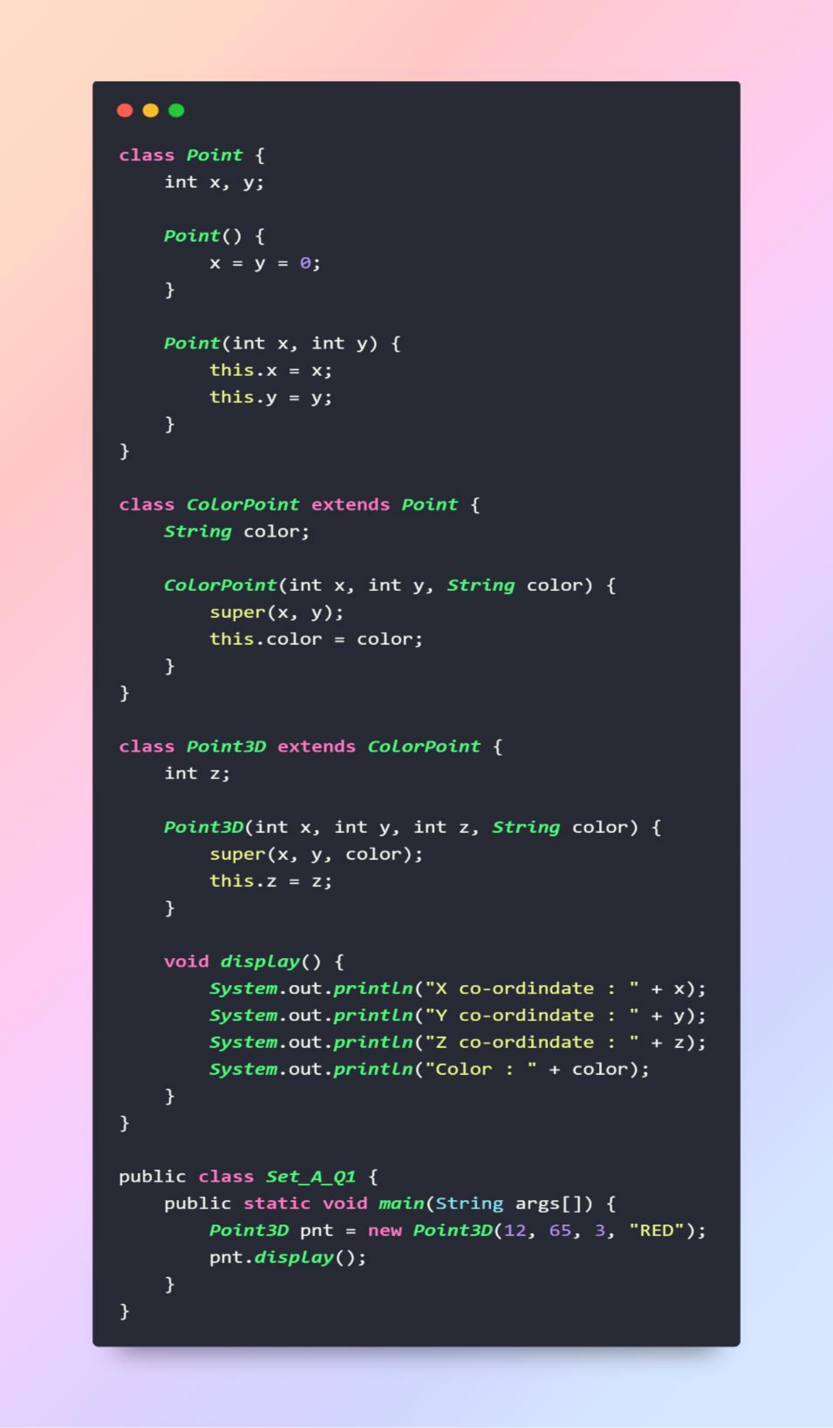


```java
import java.util.Scanner;
class Person{

    private int id;
    private static int cnt;
    public  MyDate dob;
    private String name;


    Person(){
        id++;
        id = cnt++;
        name = "Nitin";
        dob = new MyDate();
    }
    Person(int id, String name, MyDate d){
        this.id = id;
        this.name = name;
        dob.dd = d.dd;
        dob.mm = d.mm;
        dob.yy = d.yy;
        this.id = cnt++;
    }
    void accept(){

        Scanner s = new Scanner(System.in);

        System.out.print("Enter the Name of Person : ");
        name = s.next();
        System.out.println("Enter the DOB in DD-MM-YY Format : ");
        dob = dob.acceptDate();

    }
    void display(){
        System.out.println("The Id of Person is : " + id);
        System.out.println("The Name of Person is : " + name);
        System.out.println("The DOB of Person is : " + dob.display());
        System.out.println();
    }
}
public class SetC_1 {

  public static void main(String[] args) {

        Person p1 = new Person();
        p1.display();

        System.out.println("\n\n\n\n\n");

        Scanner s = new Scanner(System.in);
        System.out.println("Enter the Size of Array : ");
        int n = s.nextInt();

        Person p[] = new Person[n];

        for(int i=0;i<n;i++)
        {
            p[i] = new Person();
            System.out.println("Enter the Person Details for " + (i+1) + " : ");
            p[i].accept();
        }
        System.out.println("\n\nShowing Person Details : ");
        for(int i=0;i<n;i++){
            p[i].display();
        }
    }

}
```

Assignment 2

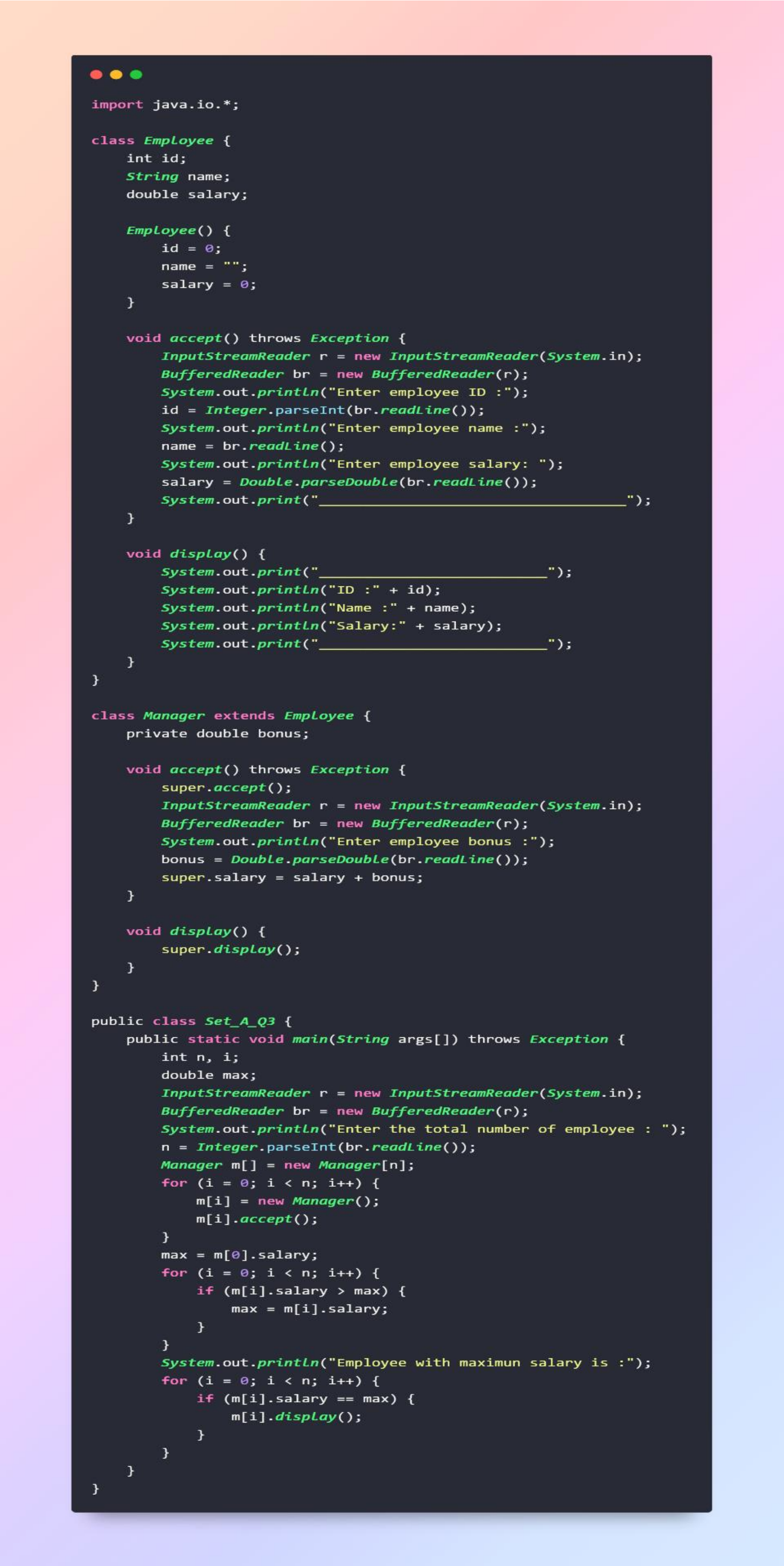
SET A

1. Define a “Point” class having members – x, y(coordinates). Define default constructor and parameterized constructors. Define two subclasses “ColourPoint” with member as colour and subclass “Point3D” with member as z (coordinate). Write display method to display the details of different types of Points.



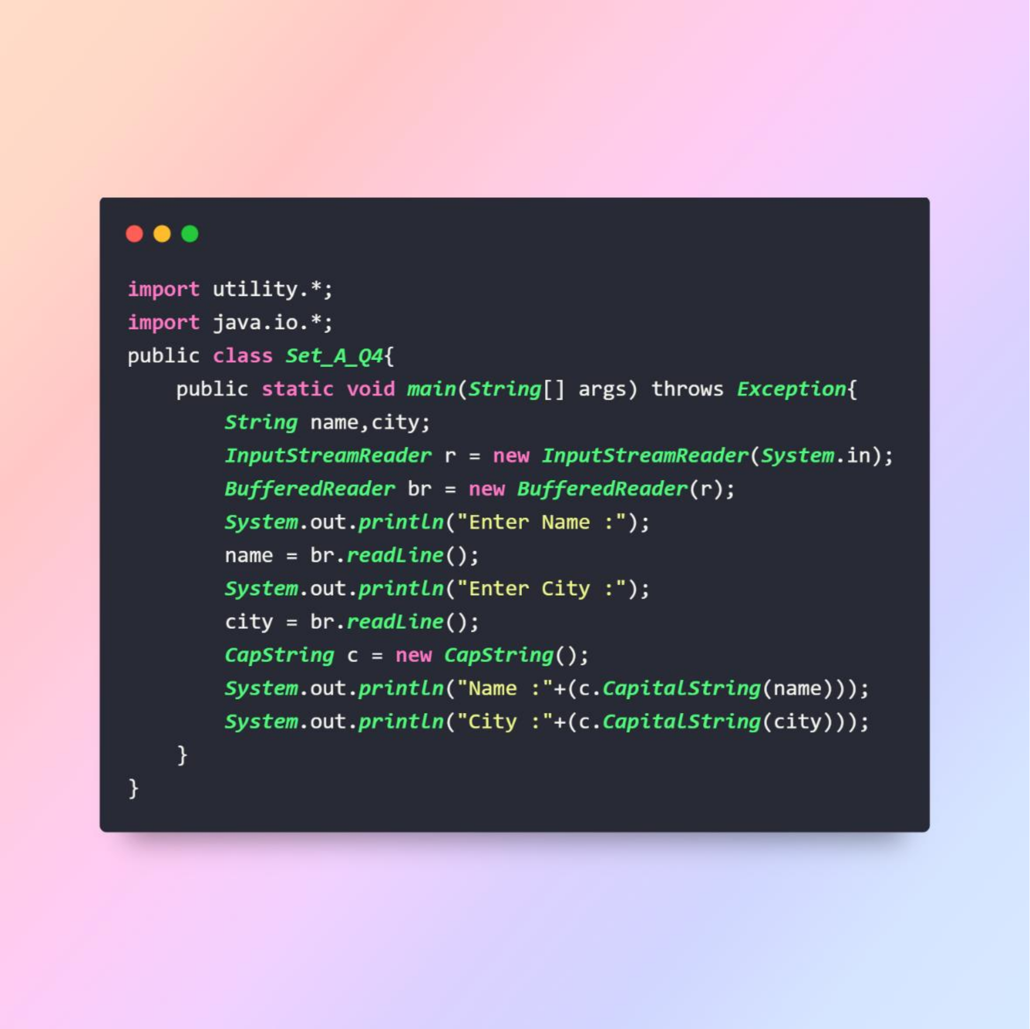
```java
class Point {
    int x, y;

    Point() {
        x = y = 0;
    }

    Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

class ColorPoint extends Point {
    String color;

    ColorPoint(int x, int y, String color) {
        super(x, y);
        this.color = color;
    }
}

class Point3D extends ColorPoint {
    int z;

    Point3D(int x, int y, int z, String color) {
        super(x, y, color);
        this.z = z;
    }

    void display() {
        System.out.println("X co-ordindate : " + x);
        System.out.println("Y co-ordindate : " + y);
        System.out.println("Z co-ordindate : " + z);
        System.out.println("Color : " + color);
    }
}

public class Set_A_Q1 {
    public static void main(String args[]) {
        Point3D pnt = new Point3D(12, 65, 3, "RED");
        pnt.display();
    }
}
```

2. Define a class Employee having members – id, name, salary. Define default constructor. Create a subclass called Manager with private member bonus. Define methods accept and display in both the classes. Create “n” objects of the Manager class and display the details of the worker having the maximum total salary (salary + bonus).



```
import java.io.*;

class Employee {
    int id;
    String name;
    double salary;

    Employee() {
        id = 0;
        name = "";
        salary = 0;
    }

    void accept() throws Exception {
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        System.out.println("Enter employee ID :");
        id = Integer.parseInt(br.readLine());
        System.out.println("Enter employee name :");
        name = br.readLine();
        System.out.println("Enter employee salary: ");
        salary = Double.parseDouble(br.readLine());
        System.out.print("__________________________________");
    }

    void display() {
        System.out.print("_________________________");
        System.out.println("ID :" + id);
        System.out.println("Name :" + name);
        System.out.println("Salary:" + salary);
        System.out.print("_________________________");
    }
}

class Manager extends Employee {
    private double bonus;

    void accept() throws Exception {
        super.accept();
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        System.out.println("Enter employee bonus :");
        bonus = Double.parseDouble(br.readLine());
        super.salary = salary + bonus;
    }

    void display() {
        super.display();
    }
}

public class Set_A_Q3 {
    public static void main(String args[]) throws Exception {
        int n, i;
        double max;
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        System.out.println("Enter the total number of employee : ");
        n = Integer.parseInt(br.readLine());
        Manager m[] = new Manager[n];
        for (i = 0; i < n; i++) {
            m[i] = new Manager();
            m[i].accept();
        }
        max = m[0].salary;
        for (i = 0; i < n; i++) {
            if (m[i].salary > max) {
                max = m[i].salary;
            }
        }
        System.out.println("Employee with maximun salary is :");
        for (i = 0; i < n; i++) {
            if (m[i].salary == max) {
                m[i].display();
            }
        }
    }
}
```

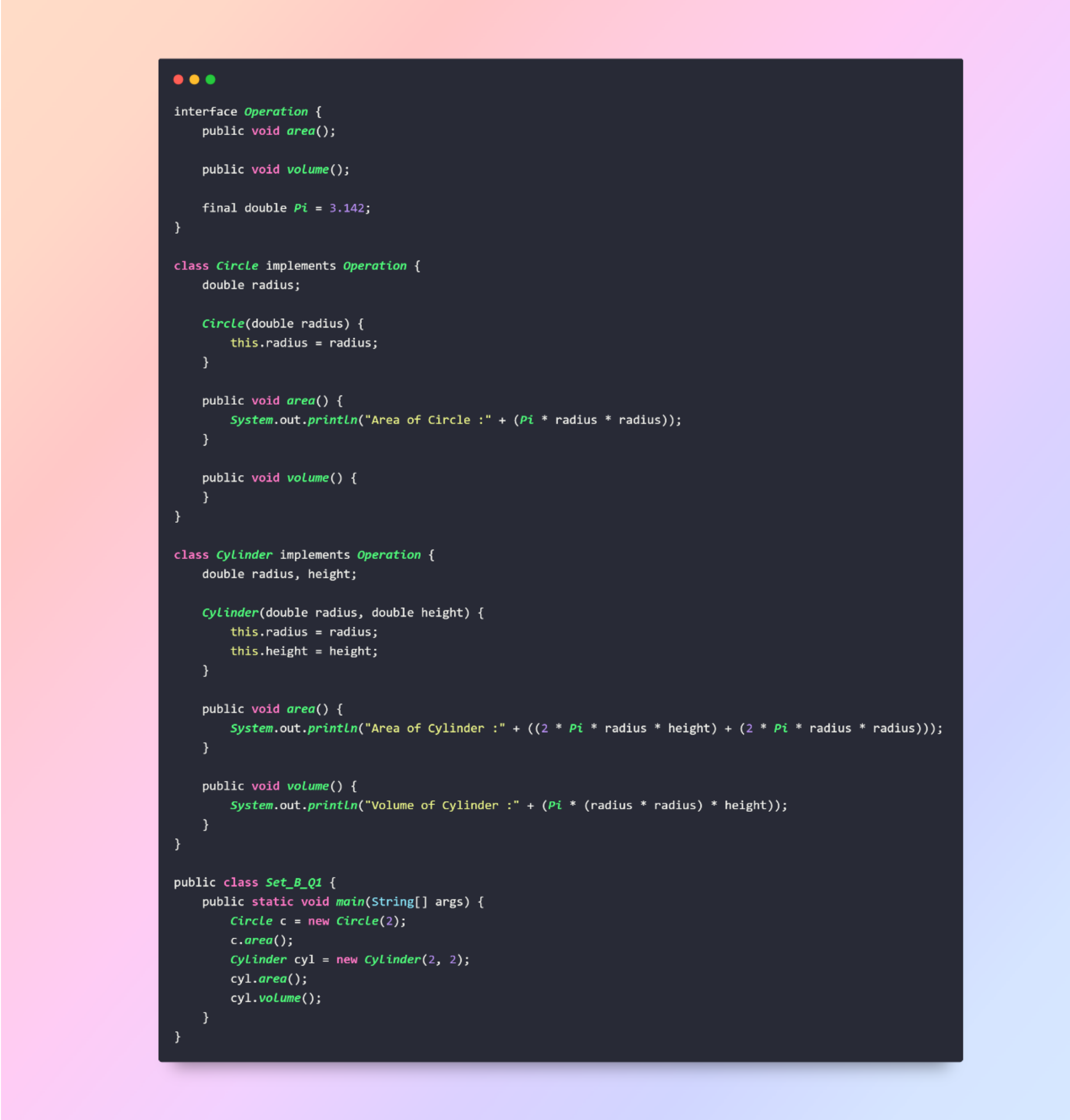
4. Create a package “utility”. Define a class CapitalString under “utility” package which will contain a method to return String with first letter capital. Create a Person class (members – name, city) outside the package. Display the person’s name with first letter as capital by making use of CapitalString.



```
import utility.*;
import java.io.*;
public class Set_A_Q4{
    public static void main(String[] args) throws Exception{
        String name,city;
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        System.out.println("Enter Name :");
        name = br.readLine();
        System.out.println("Enter City :");
        city = br.readLine();
        CapString c = new CapString();
        System.out.println("Name :"+(c.CapitalString(name)));
        System.out.println("City :"+(c.CapitalString(city)));
    }
}
```

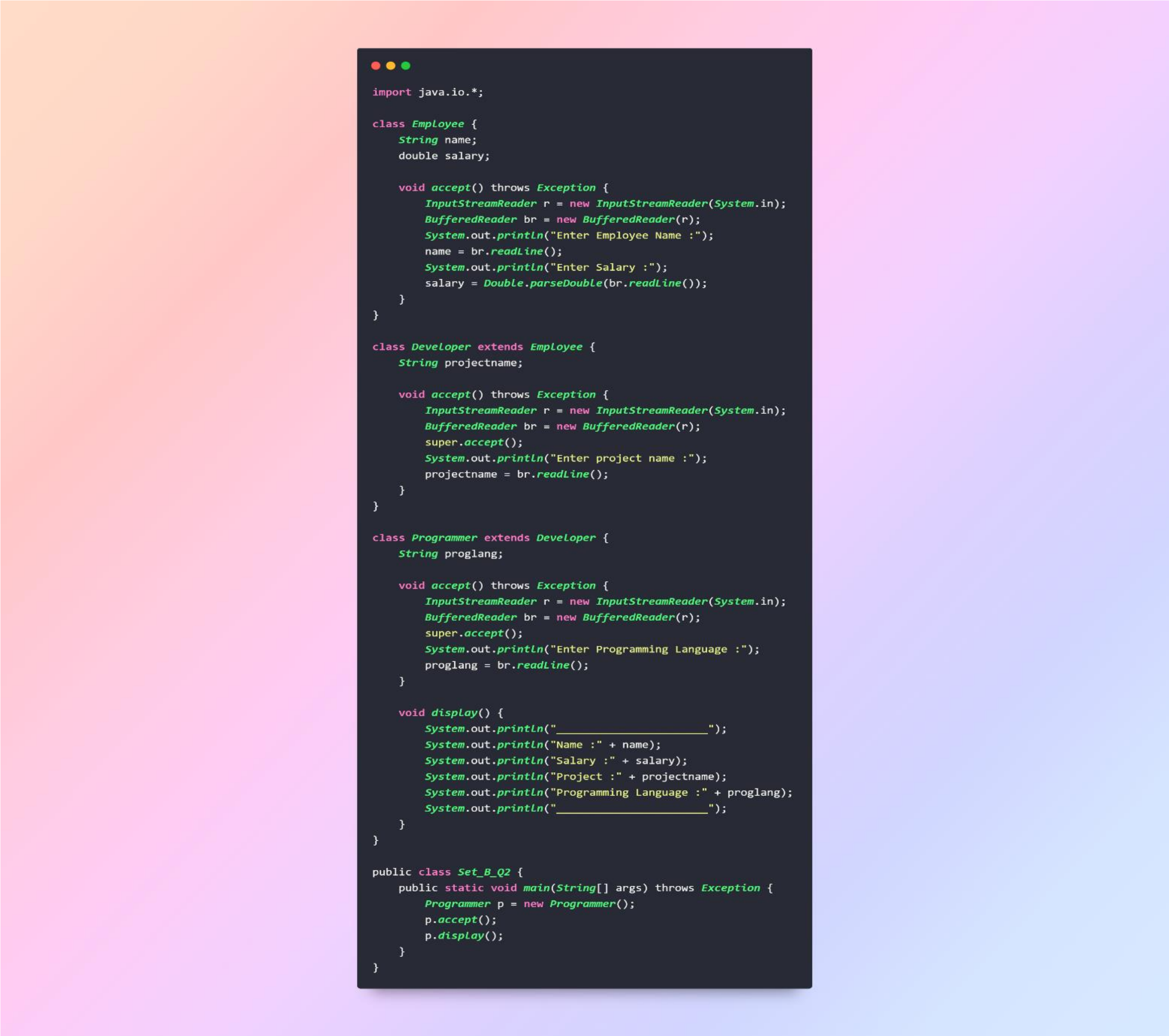
SET B

1. Define an interface “Operation” which has methods area(),volume(). Define a constant PI having a value 3.142. Create a class circle (member – radius), cylinder (members – radius, height) which implements this interface. Calculate and display the area and volume.



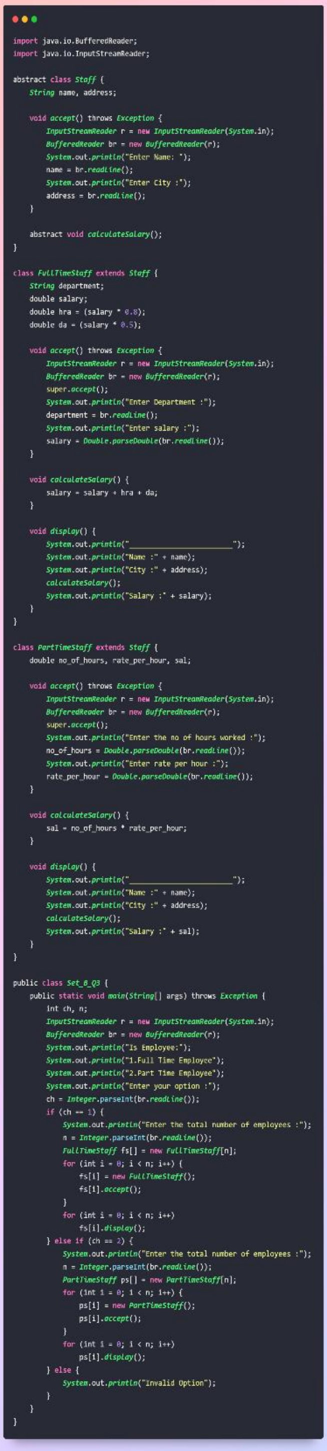
```java
interface Operation {
    public void area();

    public void volume();

    final double Pi = 3.142;
}

class Circle implements Operation {
    double radius;

    Circle(double radius) {
        this.radius = radius;
    }

    public void area() {
        System.out.println("Area of Circle :" + (Pi * radius * radius));
    }

    public void volume() {
    }
}

class Cylinder implements Operation {
    double radius, height;

    Cylinder(double radius, double height) {
        this.radius = radius;
        this.height = height;
    }

    public void area() {
        System.out.println("Area of Cylinder :" + ((2 * Pi * radius * height) + (2 * Pi * radius * radius)));
    }

    public void volume() {
        System.out.println("Volume of Cylinder :" + (Pi * (radius * radius) * height));
    }
}

public class Set_B_Q1 {
    public static void main(String[] args) {
        Circle c = new Circle(2);
        c.area();
        Cylinder cyl = new Cylinder(2, 2);
        cyl.area();
        cyl.volume();
    }
}
```

2.Write a Java program to create a super class Employee (members – name, salary). Derive a sub-class as Developer (member – projectname). Derive a sub-class Programmer (member – proglanguage) from Developer. Create object of Developer and display the details of it. Implement this multilevel inheritance with appropriate constructor and methods.s



```java
import java.io.*;

class Employee {
    String name;
    double salary;

    void accept() throws Exception {
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        System.out.println("Enter Employee Name :");
        name = br.readLine();
        System.out.println("Enter Salary :");
        salary = Double.parseDouble(br.readLine());
    }
}

class Developer extends Employee {
    String projectname;

    void accept() throws Exception {
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        super.accept();
        System.out.println("Enter project name :");
        projectname = br.readLine();
    }
}

class Programmer extends Developer {
    String proglang;

    void accept() throws Exception {
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        super.accept();
        System.out.println("Enter Programming Language :");
        proglang = br.readLine();
    }

    void display() {
        System.out.println("_______________________");
        System.out.println("Name :" + name);
        System.out.println("Salary :" + salary);
        System.out.println("Project :" + projectname);
        System.out.println("Programming Language :" + proglang);
        System.out.println("_______________________");
    }
}

public class Set_B_Q2 {
    public static void main(String[] args) throws Exception {
        Programmer p = new Programmer();
        p.accept();
        p.display();
    }
}
```

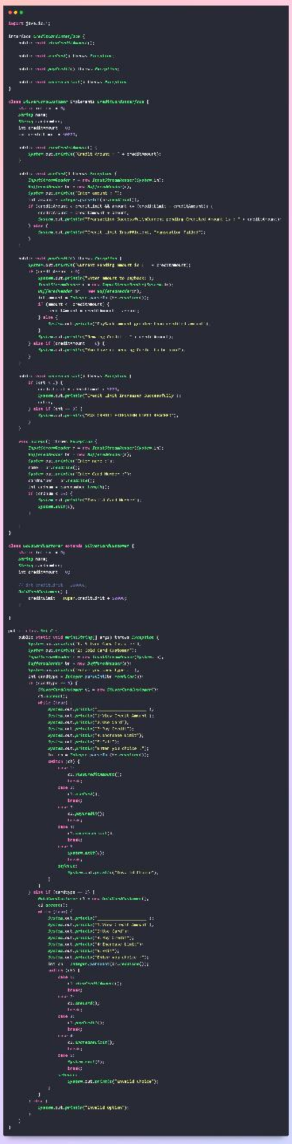
3.Define an abstract class Staff with members name and address. Define two sub-classes of this class – FullTimeStaff (members - department, salary, hra - 8% of salary, da – 5% of salary) and PartTimeStaff (members - number-of-hours, rate-per-hour). Define appropriate constructors. Write abstract method as calculateSalary() in Staff class. Implement this method in subclasses. Create n objects which could be of either FullTimeStaff or PartTimeStaff class by asking the user‘s choice. Display details of all FullTimeStaff objects and all PartTimeStaff objects along with their salary.



```java
import java.io.BufferedReader;
import java.io.InputStreamReader;

abstract class Staff {
    String name, address;

    void accept() throws Exception {
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        System.out.println("Enter Name: ");
        name = br.readLine();
        System.out.println("Enter City :");
        address = br.readLine();
    }

    abstract void calculateSalary();
}

class FullTimeStaff extends Staff {
    String department;
    double salary;
    double hra = (salary * 0.8);
    double da = (salary * 0.5);

    void accept() throws Exception {
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        super.accept();
        System.out.println("Enter Department :");
        department = br.readLine();
        System.out.println("Enter salary :");
        salary = Double.parseDouble(br.readLine());
    }

    void calculateSalary() {
        salary = salary + hra + da;
    }

    void display() {
        System.out.println("_________________________");
        System.out.println("Name :" + name);
        System.out.println("City :" + address);
        calculateSalary();
        System.out.println("Salary :" + salary);
    }
}

class PartTimeStaff extends Staff {
    double no_of_hours, rate_per_hour, sal;

    void accept() throws Exception {
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        super.accept();
        System.out.println("Enter the no of hours worked :");
        no_of_hours = Double.parseDouble(br.readLine());
        System.out.println("Enter rate per hour :");
        rate_per_hour = Double.parseDouble(br.readLine());
    }

    void calculateSalary() {
        sal = no_of_hours * rate_per_hour;
    }

    void display() {
        System.out.println("_________________________");
        System.out.println("Name :" + name);
        System.out.println("City :" + address);
        calculateSalary();
        System.out.println("Salary :" + sal);
    }
}

public class Set_B_Q3 {
    public static void main(String[] args) throws Exception {
        int ch, n;
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        System.out.println("Is Employee:");
        System.out.println("1.Full Time Employee");
        System.out.println("2.Part Time Employee");
        System.out.println("Enter your option :");
        ch = Integer.parseInt(br.readLine());
        if (ch == 1) {
            System.out.println("Enter the total number of employees :");
            n = Integer.parseInt(br.readLine());
            FullTimeStaff fs[] = new FullTimeStaff[n];
            for (int i = 0; i < n; i++) {
                fs[i] = new FullTimeStaff();
                fs[i].accept();
            }
            for (int i = 0; i < n; i++)
                fs[i].display();
        } else if (ch == 2) {
            System.out.println("Enter the total number of employees :");
            n = Integer.parseInt(br.readLine());
            PartTimeStaff ps[] = new PartTimeStaff[n];
            for (int i = 0; i < n; i++) {
                ps[i] = new PartTimeStaff();
                ps[i].accept();
            }
            for (int i = 0; i < n; i++)
                ps[i].display();
        } else {
            System.out.println("Invalid Option");
        }
    }
}
```

SET C

1. Create an interface — CreditCardInterface with method: viewCreditAmount(), useCard(), payCredit() and increaseLimit(). Create a class SilverCardCustomer (name, cardnumber (16digits), creditAmount – initialized to 0, creditLimit - set to 50,000) which implements the above interface. Inherit class GoldCardCustomer from SilverCardCustomer having the same methods but creditLimit of 1,00,000. Create an object of each class and perform operations. Display appropriate messages for success or failure of transactions. (Use method overriding) i. useCard() method increases the creditAmount by a specific amount upto creditLimit ii. payCredit() reduces the creditAmount by a specific amount. iii. increaseLimit() increases the creditLimit for GoldCardCustomers (only 3 times, not more than 5000Rs. each time

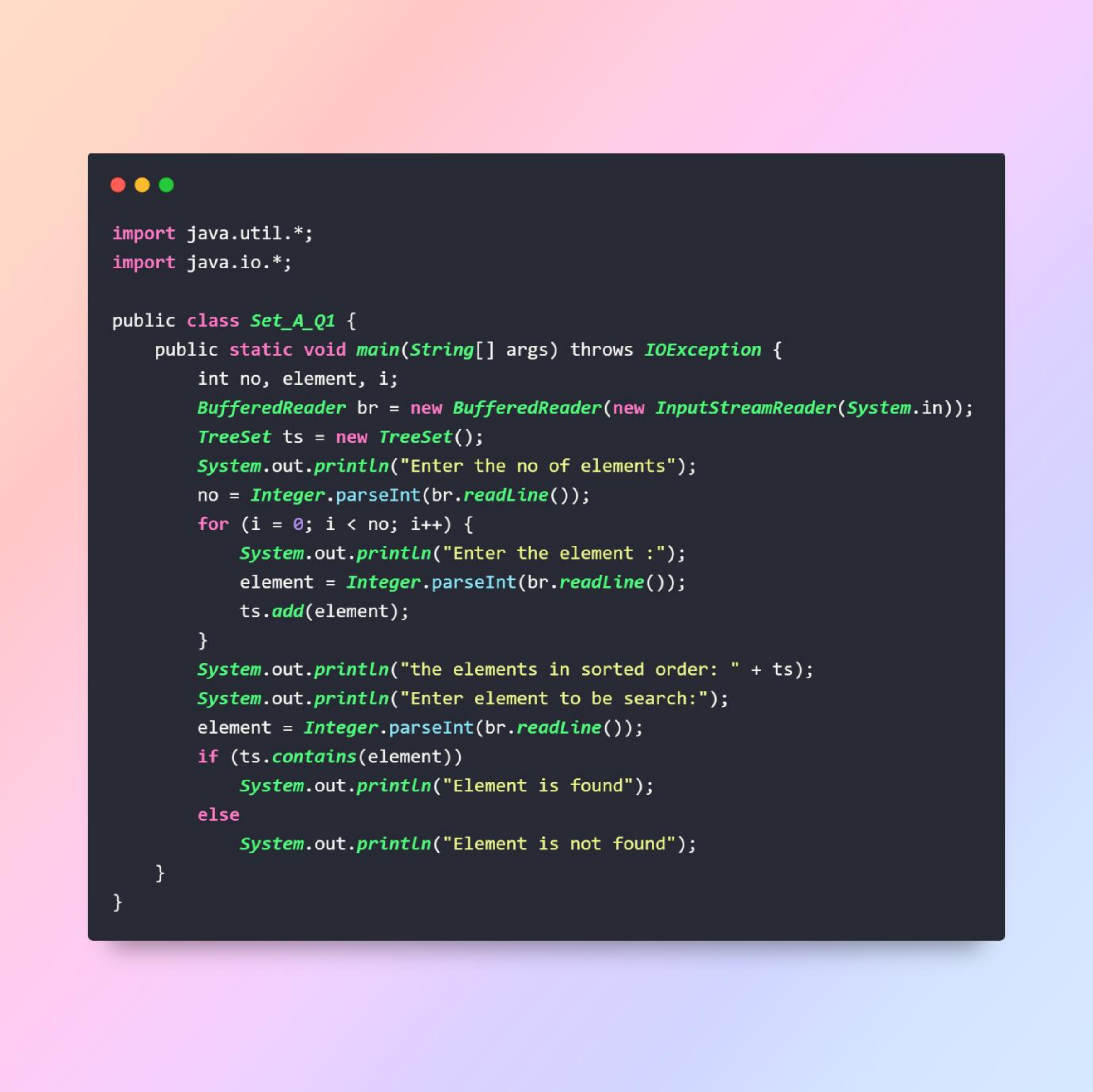


```
import java.io.*;

interface CreditCardInterface {
    public void viewCreditAmount();

    public void useCard() throws Exception;

    public void payCredit() throws Exception;

    public void increaseLimit() throws Exception;
}

class SilverCardCustomer implements CreditCardInterface {
    static int cnt = 0;
    String name;
    String cardnumber;
    int creditAmount = 0;
    int creditLimit = 50000;

    public void viewCreditAmount() {
        System.out.println("Credit Amount : " + creditAmount);
    }

    public void useCard() throws Exception {
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        System.out.println("Enter amount : ");
        int amount = Integer.parseInt(br.readLine());
        if (creditAmount < creditLimit && amount <= (creditLimit - creditAmount)) {
            creditAmount = creditAmount + amount;
            System.out.println("Transaction Successful\nCurrent pending Credited Amount is : " + creditAmount);
        } else {
            System.out.println("Credit Limit Insufficient, Transaction Failed");
        }
    }

    public void payCredit() throws Exception {
        System.out.println("Current Pending Amount is : " + creditAmount);
        if (creditAmount > 0) {
            System.out.println("Enter amount to payback: ");
            InputStreamReader r = new InputStreamReader(System.in);
            BufferedReader br = new BufferedReader(r);
            int amount = Integer.parseInt(br.readLine());
            if (amount <= creditAmount) {
                creditAmount = creditAmount - amount;
            } else {
                System.out.println("Payback amount greater than credited amount");
            }
            System.out.println("Remaining Credit : " + creditAmount);
        } else if (creditAmount == 0) {
            System.out.println("You have no pending Credit to be paid");
        }
    }

    public void increaseLimit() throws Exception {
        if (cnt < 3) {
            creditLimit = creditLimit + 5000;
            System.out.println("Credit Limit Increased Successfully ");
            cnt++;
        } else if (cnt == 3) {
            System.out.println("MAX CREDIT INCREASE LIMIT REACHED");
        }
    }

    void accept() throws Exception {
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        System.out.println("Enter name : ");
        name = br.readLine();
        System.out.println("Enter Card Number : ");
        cardnumber = br.readLine();
        int cardnum = cardnumber.length();
        if (cardnum < 16) {
            System.out.println("Invalid Card Number");
            System.exit(0);
        }

    }
}

class GoldCardCustomer extends SilverCardCustomer {
    static int cnt = 0;
    String name;
    String cardnumber;
    int creditAmount = 0;

    // int creditLimit = 100000;
    GoldCardCustomer() {
        creditLimit = super.creditLimit + 50000;
    }

}

public class [illegible] {
    public static void main(String[] args) throws Exception {
        System.out.println("1: Silver Card Customer");
        System.out.println("2: Gold Card Customer");
        InputStreamReader r = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(r);
        System.out.println("Enter your card type : ");
        int cardtype = Integer.parseInt(br.readLine());
        if (cardtype == 1) {
            SilverCardCustomer c1 = new SilverCardCustomer();
            c1.accept();
            while (true) {
                System.out.println("____________________");
                System.out.println("1:View Credit Amount");
                System.out.println("2:Use Card");
                System.out.println("3:Pay Credit");
                System.out.println("4:Increase Limit");
                System.out.println("5:Exit");
                System.out.println("Enter your choice :");
                int ch = Integer.parseInt(br.readLine());
                switch (ch) {
                    case 1:
                        c1.viewCreditAmount();
                        break;
                    case 2:
                        c1.useCard();
                        break;
                    case 3:
                        c1.payCredit();
                        break;
                    case 4:
                        c1.increaseLimit();
                        break;
                    case 5:
                        System.exit(0);
                        break;
                    default:
                        System.out.println("Invalid Choice");
                }
            }
        } else if (cardtype == 2) {
            GoldCardCustomer c2 = new GoldCardCustomer();
            c2.accept();
            while (true) {
                System.out.println("____________________");
                System.out.println("1:View Credit Amount");
                System.out.println("2:Use Card");
                System.out.println("3:Pay Credit");
                System.out.println("4:Increase Limit");
                System.out.println("5:Exit");
                System.out.println("Enter your choice :");
                int ch = Integer.parseInt(br.readLine());
                switch (ch) {
                    case 1:
                        c2.viewCreditAmount();
                        break;
                    case 2:
                        c2.useCard();
                        break;
                    case 3:
                        c2.payCredit();
                        break;
                    case 4:
                        c2.increaseLimit();
                        break;
                    case 5:
                        System.exit(0);
                        break;
                    default:
                        System.out.println("Invalid Choice");
                }
            }
        } else {
            System.out.println("Invalid Option");
        }
    }
}
```

Assignment 3

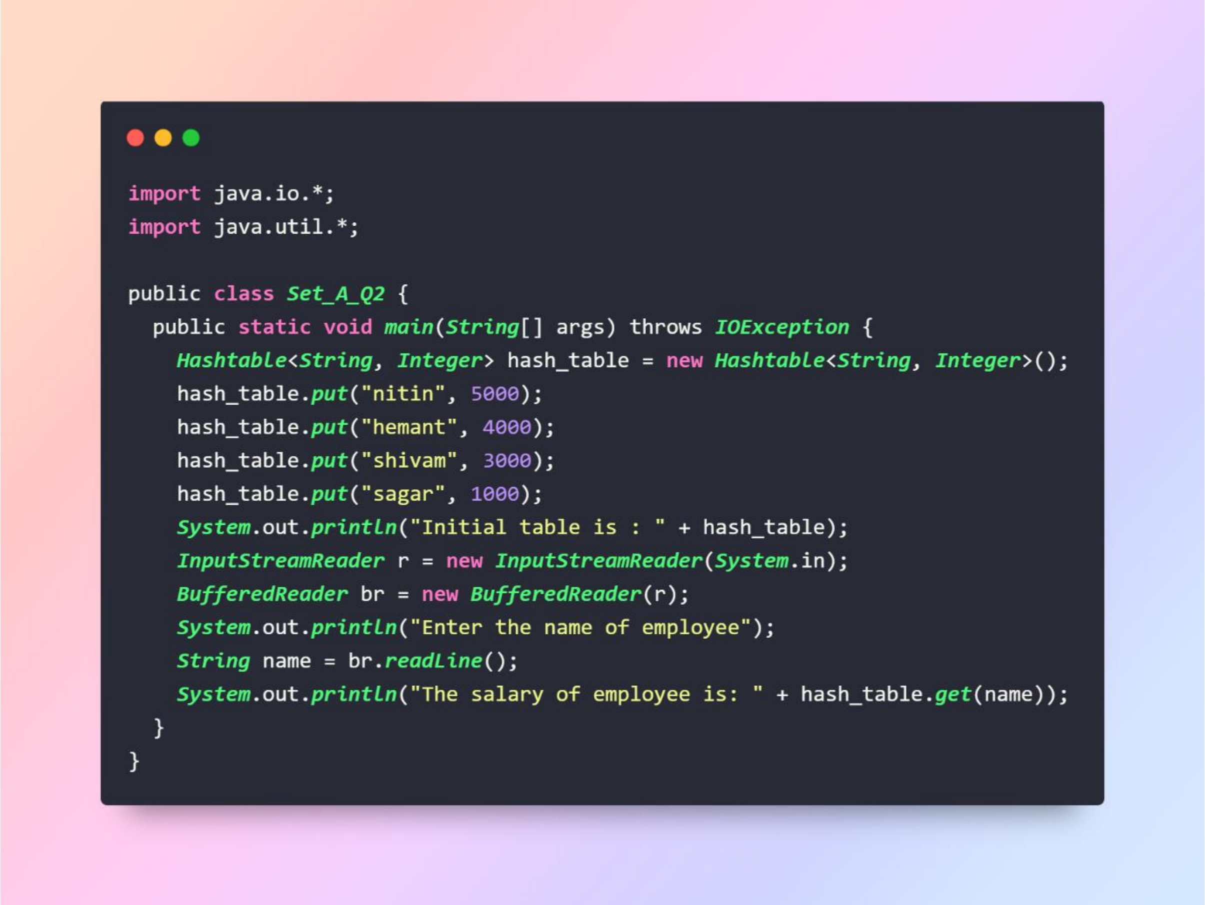
SET A

1. Accept n integers from the user and store them in a collection. Display them in the sorted order. The collection should not accept duplicate elements. (Use a suitable collection). Search for a particular element using predefined search method in the Collection framework.



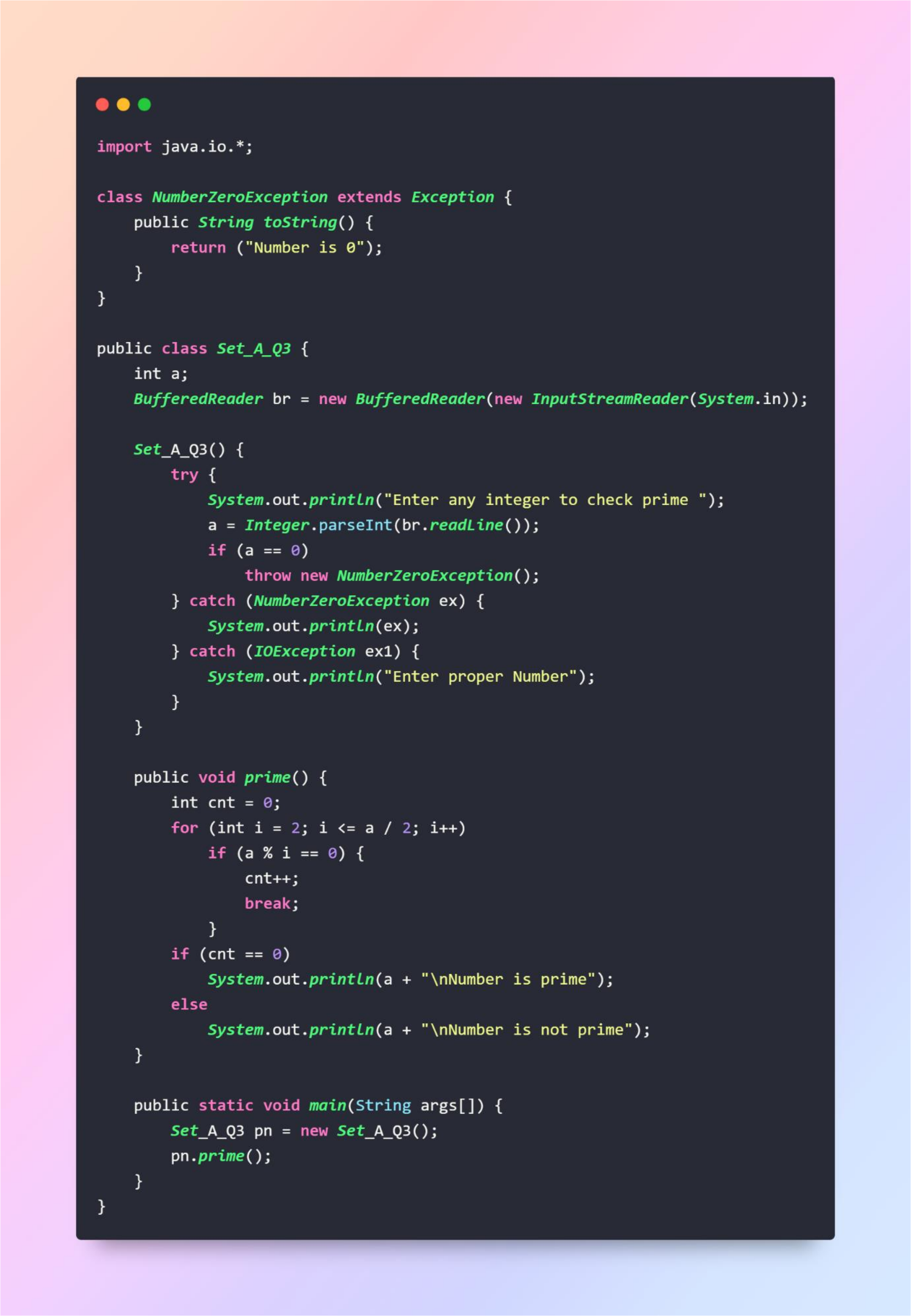
```
import java.util.*;
import java.io.*;

public class Set_A_Q1 {
    public static void main(String[] args) throws IOException {
        int no, element, i;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        TreeSet ts = new TreeSet();
        System.out.println("Enter the no of elements");
        no = Integer.parseInt(br.readLine());
        for (i = 0; i < no; i++) {
            System.out.println("Enter the element :");
            element = Integer.parseInt(br.readLine());
            ts.add(element);
        }
        System.out.println("the elements in sorted order: " + ts);
        System.out.println("Enter element to be search:");
        element = Integer.parseInt(br.readLine());
        if (ts.contains(element))
            System.out.println("Element is found");
        else
            System.out.println("Element is not found");
    }
}
```

2. Create a Hash table containing Employee name and Salary. Display the details of the hash table. Also search for a specific Employee and display Salary of that Employee.



```java
import java.io.*;
import java.util.*;

public class Set_A_Q2 {
  public static void main(String[] args) throws IOException {
    Hashtable<String, Integer> hash_table = new Hashtable<String, Integer>();
    hash_table.put("nitin", 5000);
    hash_table.put("hemant", 4000);
    hash_table.put("shivam", 3000);
    hash_table.put("sagar", 1000);
    System.out.println("Initial table is : " + hash_table);
    InputStreamReader r = new InputStreamReader(System.in);
    BufferedReader br = new BufferedReader(r);
    System.out.println("Enter the name of employee");
    String name = br.readLine();
    System.out.println("The salary of employee is: " + hash_table.get(name));
  }
}
```

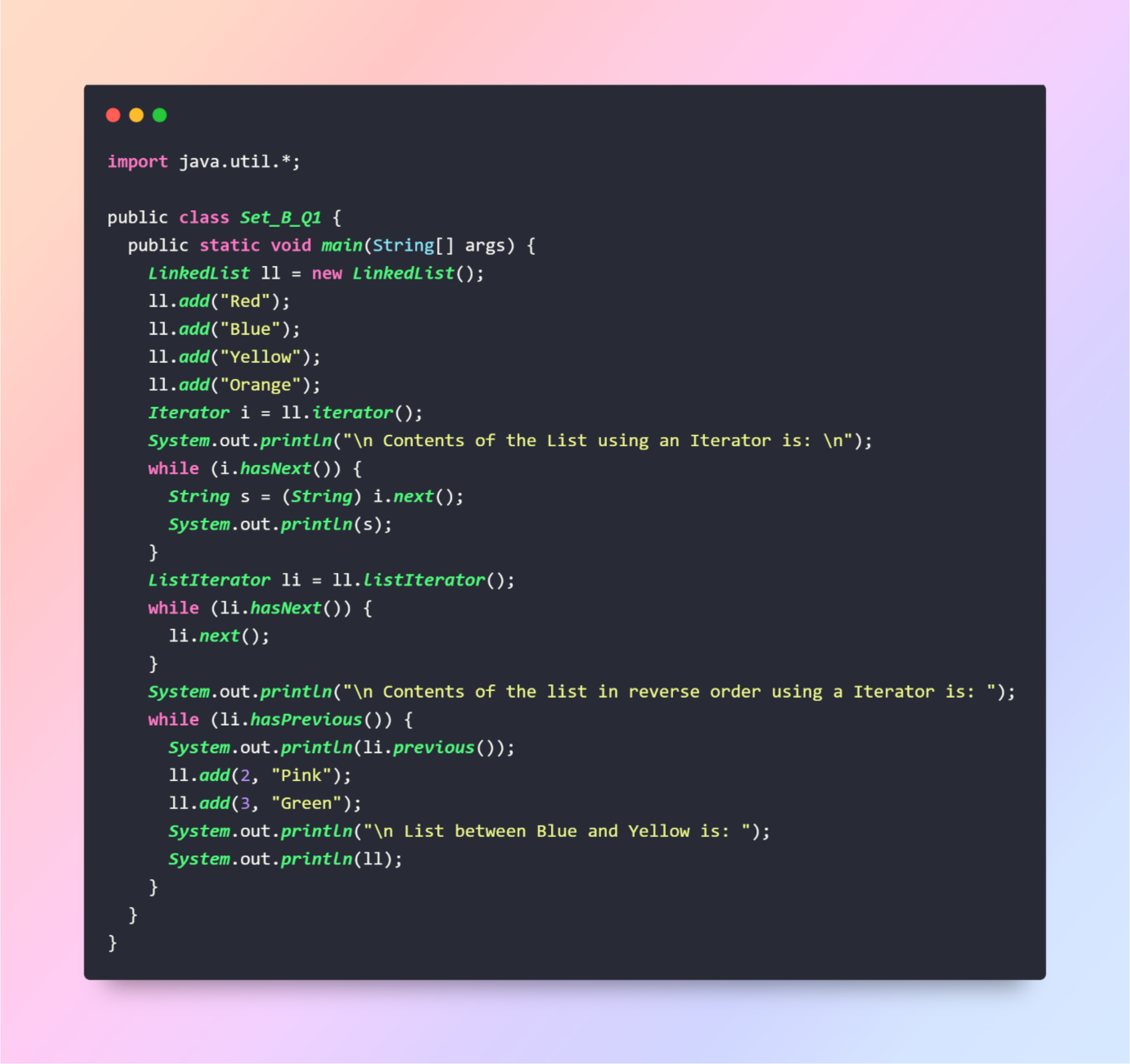
3. Write a java program to accept a number from the user, if number is zero then throw user defined exception —Number is 0, otherwise check whether no is prime or not. 4. Write a java program that displays the number of characters, lines and words of a file.



```java
import java.io.*;

class NumberZeroException extends Exception {
    public String toString() {
        return ("Number is 0");
    }
}

public class Set_A_Q3 {
    int a;
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

    Set_A_Q3() {
        try {
            System.out.println("Enter any integer to check prime ");
            a = Integer.parseInt(br.readLine());
            if (a == 0)
                throw new NumberZeroException();
        } catch (NumberZeroException ex) {
            System.out.println(ex);
        } catch (IOException ex1) {
            System.out.println("Enter proper Number");
        }
    }

    public void prime() {
        int cnt = 0;
        for (int i = 2; i <= a / 2; i++)
            if (a % i == 0) {
                cnt++;
                break;
            }
        if (cnt == 0)
            System.out.println(a + "\nNumber is prime");
        else
            System.out.println(a + "\nNumber is not prime");
    }

    public static void main(String args[]) {
        Set_A_Q3 pn = new Set_A_Q3();
        pn.prime();
    }
}
```

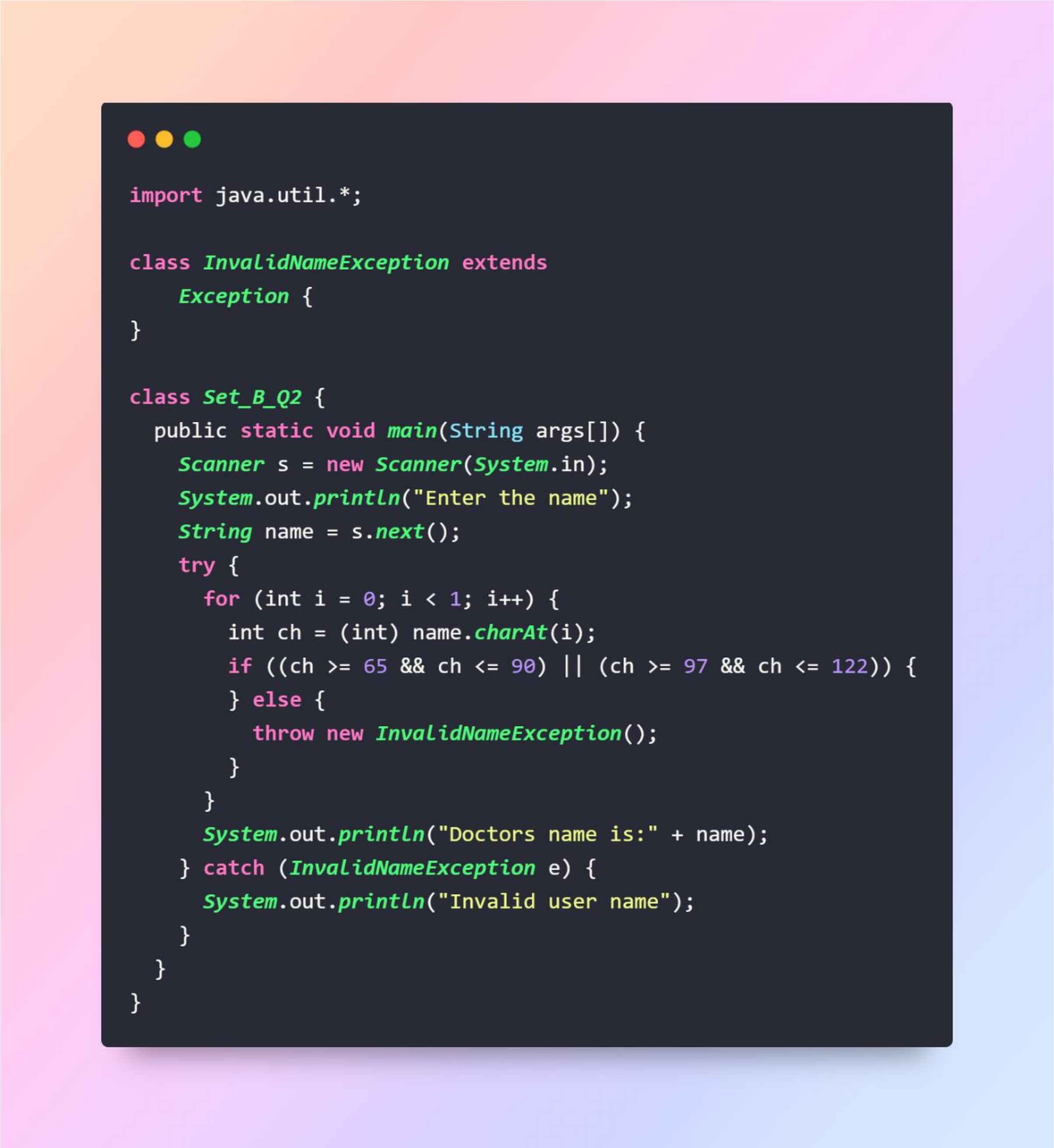
SET B

1. Construct a linked List containing names of colours: red, blue, yellow and orange. Then extend your program to do the following: i. Display the contents of the List using an Iterator ii. Display the contents of the List in reverse order using a ListIterator iii. Create another list containing pink and green. Insert the elements of this list between blue and yellow.



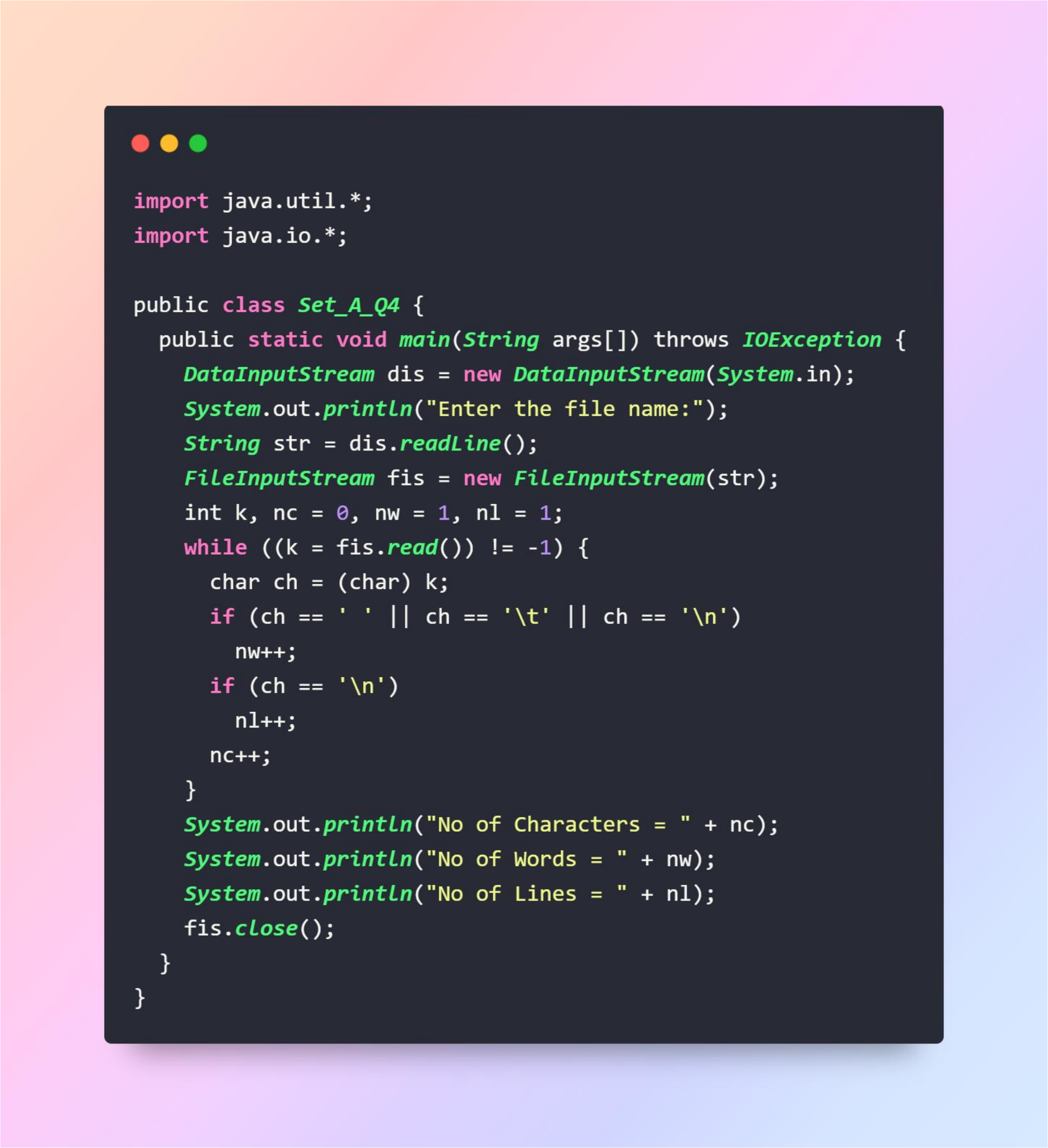
```java
import java.util.*;

public class Set_B_Q1 {
  public static void main(String[] args) {
    LinkedList ll = new LinkedList();
    ll.add("Red");
    ll.add("Blue");
    ll.add("Yellow");
    ll.add("Orange");
    Iterator i = ll.iterator();
    System.out.println("\n Contents of the List using an Iterator is: \n");
    while (i.hasNext()) {
      String s = (String) i.next();
      System.out.println(s);
    }
    ListIterator li = ll.listIterator();
    while (li.hasNext()) {
      li.next();
    }
    System.out.println("\n Contents of the list in reverse order using a Iterator is: ");
    while (li.hasPrevious()) {
      System.out.println(li.previous());
      ll.add(2, "Pink");
      ll.add(3, "Green");
      System.out.println("\n List between Blue and Yellow is: ");
      System.out.println(ll);
    }
  }
}
```

2. Write a java program to accept Doctor Name from the user and check whether it is valid or not. (It should not contain digits and special symbol) If it is not valid then throw user defined Exception - Name is Invalid -- otherwise display it.



```java
import java.util.*;

class InvalidNameException extends
    Exception {
}

class Set_B_Q2 {
  public static void main(String args[]) {
    Scanner s = new Scanner(System.in);
    System.out.println("Enter the name");
    String name = s.next();
    try {
      for (int i = 0; i < 1; i++) {
        int ch = (int) name.charAt(i);
        if ((ch >= 65 && ch <= 90) || (ch >= 97 && ch <= 122)) {
        } else {
          throw new InvalidNameException();
        }
      }
      System.out.println("Doctors name is:" + name);
    } catch (InvalidNameException e) {
      System.out.println("Invalid user name");
    }
  }
}
```

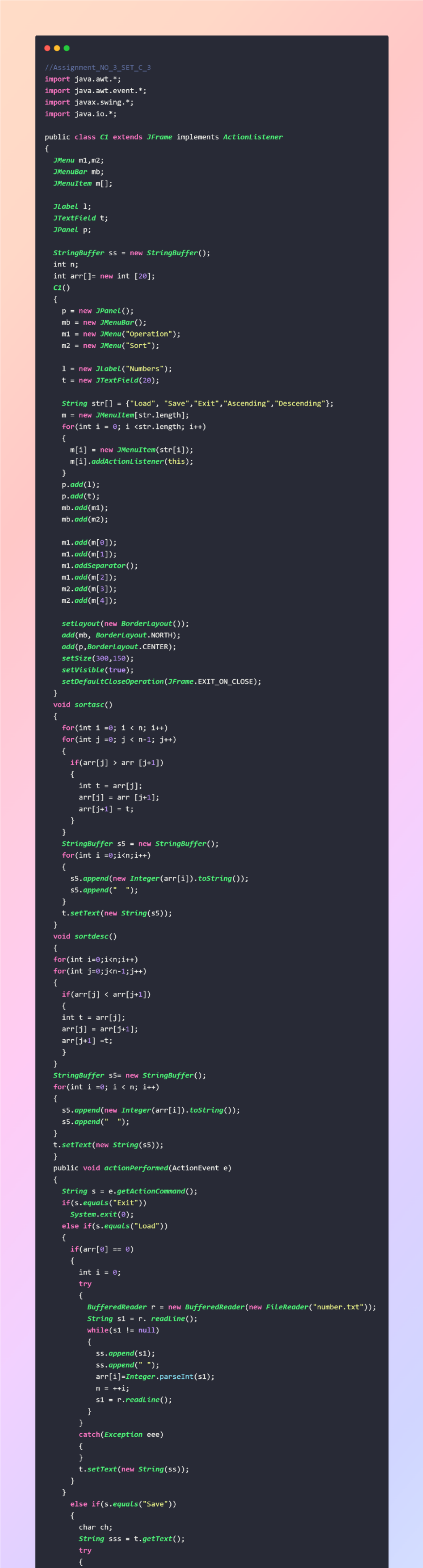
3. Write a java program to accept details of n customers (c_id, cname, address, mobile_no) from user and store it in a file (Use DataOutputStream class). Display the details of customers by reading it from file.(Use DataInputStream class)



```java
import java.util.*;
import java.io.*;

public class Set_A_Q4 {
  public static void main(String args[]) throws IOException {
    DataInputStream dis = new DataInputStream(System.in);
    System.out.println("Enter the file name:");
    String str = dis.readLine();
    FileInputStream fis = new FileInputStream(str);
    int k, nc = 0, nw = 1, nl = 1;
    while ((k = fis.read()) != -1) {
      char ch = (char) k;
      if (ch == ' ' || ch == '\t' || ch == '\n')
        nw++;
      if (ch == '\n')
        nl++;
      nc++;
    }
    System.out.println("No of Characters = " + nc);
    System.out.println("No of Words = " + nw);
    System.out.println("No of Lines = " + nl);
    fis.close();
  }
}
```

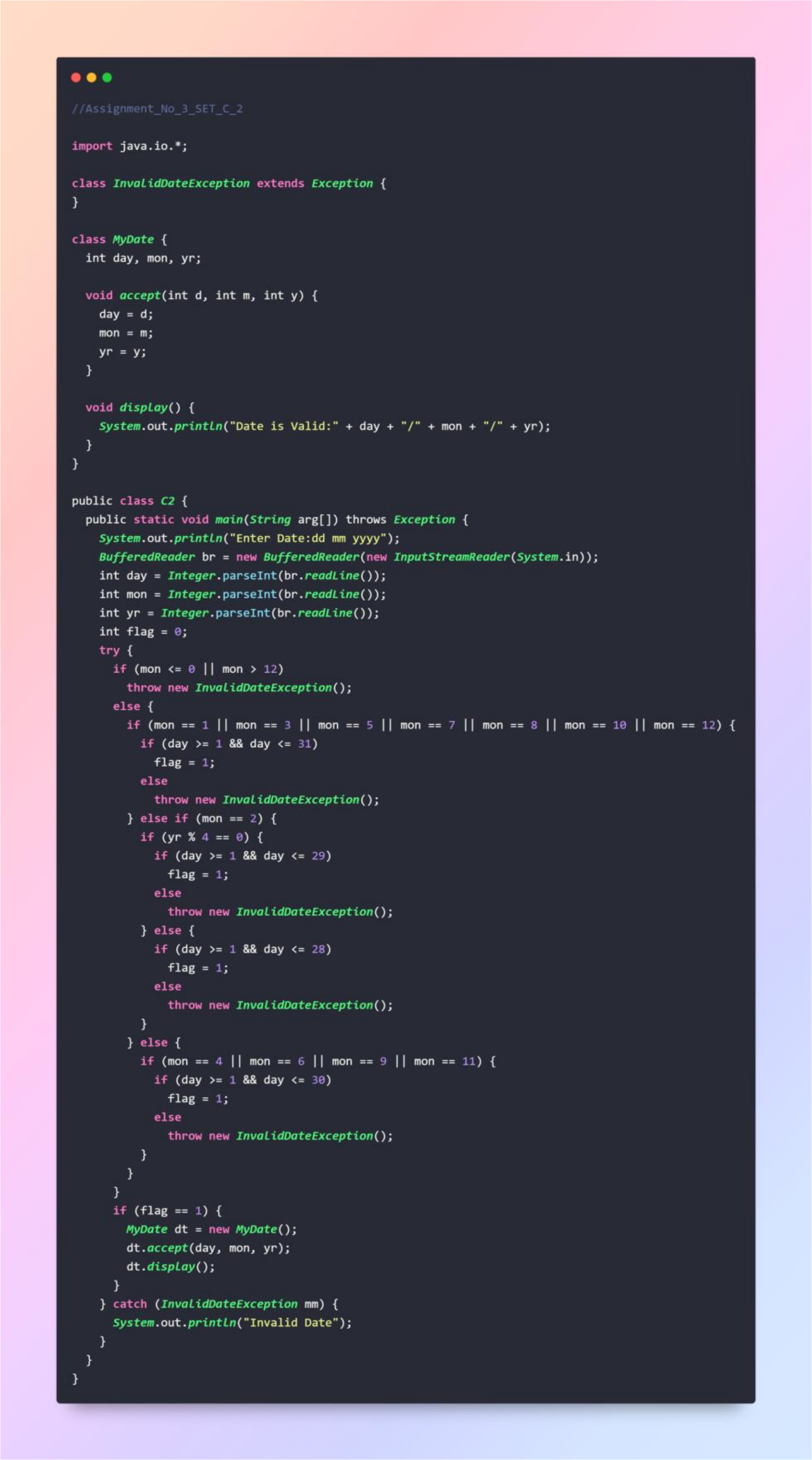
SET C

1. Write a menu driven program to perform the following operations on a set of integers. 1. Load: This operation should generate 10 random integers (2 digit) and display the number on screen. 2. Save: The save operation should save the number to a file “number.txt”. 3. Exit



```
//Assignment_NO_3_SET_C_3
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class C1 extends JFrame implements ActionListener
{
  JMenu m1,m2;
  JMenuBar mb;
  JMenuItem m[];

  JLabel l;
  JTextField t;
  JPanel p;

  StringBuffer ss = new StringBuffer();
  int n;
  int arr[]= new int [20];
  C1()
  {
    p = new JPanel();
    mb = new JMenuBar();
    m1 = new JMenu("Operation");
    m2 = new JMenu("Sort");

    l = new JLabel("Numbers");
    t = new JTextField(20);

    String str[] = {"Load", "Save","Exit","Ascending","Descending"};
    m = new JMenuItem[str.length];
    for(int i = 0; i <str.length; i++)
    {
      m[i] = new JMenuItem(str[i]);
      m[i].addActionListener(this);
    }
    p.add(l);
    p.add(t);
    mb.add(m1);
    mb.add(m2);

    m1.add(m[0]);
    m1.add(m[1]);
    m1.addSeparator();
    m1.add(m[2]);
    m2.add(m[3]);
    m2.add(m[4]);

    setLayout(new BorderLayout());
    add(mb, BorderLayout.NORTH);
    add(p,BorderLayout.CENTER);
    setSize(300,150);
    setVisible(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  }
  void sortasc()
  {
    for(int i =0; i < n; i++)
    for(int j =0; j < n-1; j++)
    {
      if(arr[j] > arr [j+1])
      {
        int t = arr[j];
        arr[j] = arr [j+1];
        arr[j+1] = t;
      }
    }
    StringBuffer s5 = new StringBuffer();
    for(int i =0;i<n;i++)
    {
      s5.append(new Integer(arr[i]).toString());
      s5.append("  ");
    }
    t.setText(new String(s5));
  }
  void sortdesc()
  {
  for(int i=0;i<n;i++)
  for(int j=0;j<n-1;j++)
  {
    if(arr[j] < arr[j+1])
    {
    int t = arr[j];
    arr[j] = arr[j+1];
    arr[j+1] =t;
    }
  }
  StringBuffer s5= new StringBuffer();
  for(int i =0; i < n; i++)
  {
    s5.append(new Integer(arr[i]).toString());
    s5.append("  ");
  }
  t.setText(new String(s5));
  }
  public void actionPerformed(ActionEvent e)
  {
    String s = e.getActionCommand();
    if(s.equals("Exit"))
      System.exit(0);
    else if(s.equals("Load"))
    {
      if(arr[0] == 0)
      {
        int i = 0;
        try
        {
          BufferedReader r = new BufferedReader(new FileReader("number.txt"));
          String s1 = r. readLine();
          while(s1 != null)
          {
            ss.append(s1);
            ss.append(" ");
            arr[i]=Integer.parseInt(s1);
            n = ++i;
            s1 = r.readLine();
          }
        }
        catch(Exception eee)
        {
        }
        t.setText(new String(ss));
      }
    }
      else if(s.equals("Save"))
      {
        char ch;
        String sss = t.getText();
        try
        {
```

2. Define a class MyDate (day, month, year) with methods to accept and display a MyDate object. Accept date as dd, mm, yyyy. Throw user defined exception “InvalidDateException” if the date is invalid. Examples of invalid dates : 12 15 2015, 31 6 1990, 29 2 2001

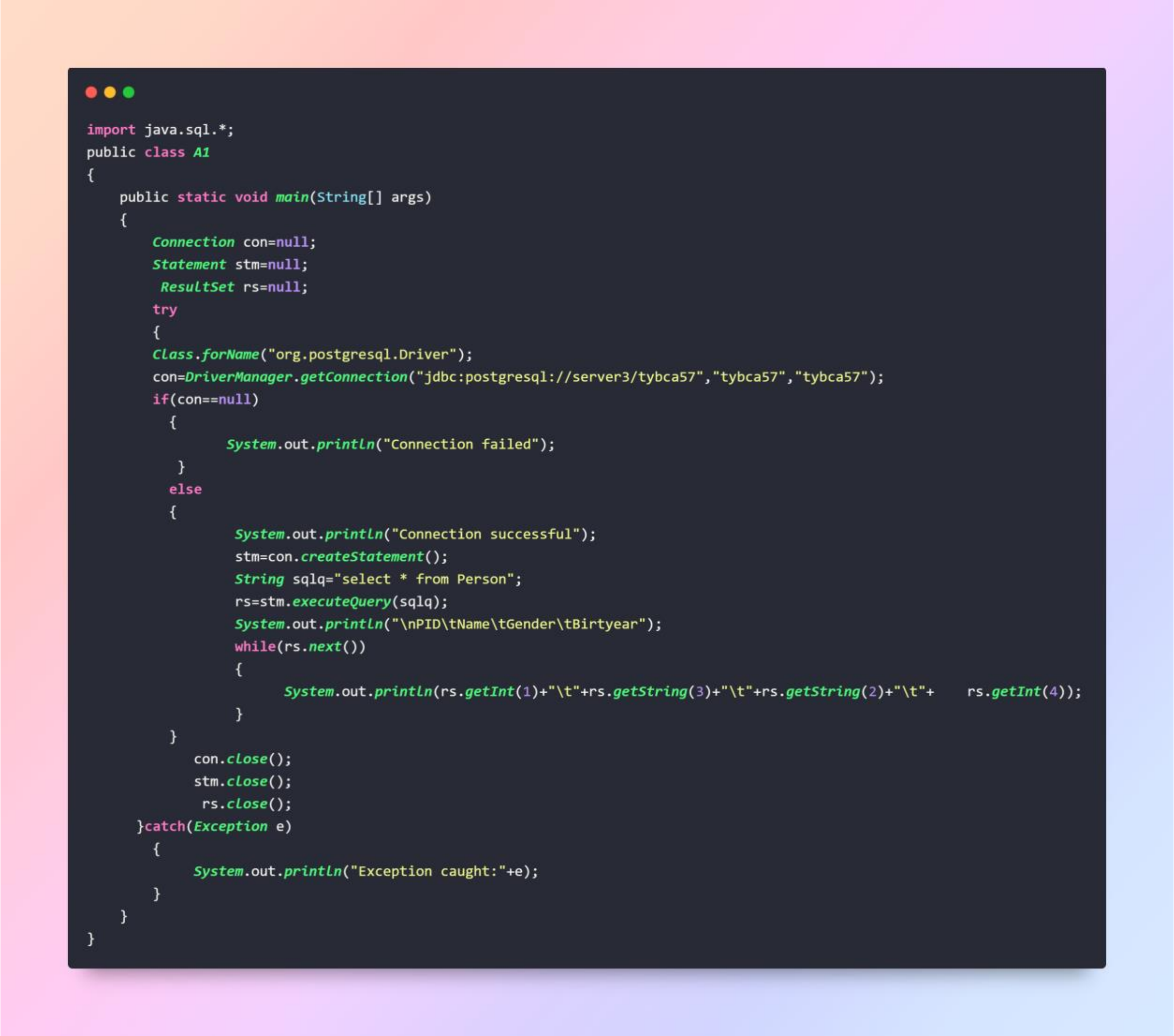


```java
//Assignment_No_3_SET_C_2

import java.io.*;

class InvalidDateException extends Exception {
}

class MyDate {
  int day, mon, yr;

  void accept(int d, int m, int y) {
    day = d;
    mon = m;
    yr = y;
  }

  void display() {
    System.out.println("Date is Valid:" + day + "/" + mon + "/" + yr);
  }
}

public class C2 {
  public static void main(String arg[]) throws Exception {
    System.out.println("Enter Date:dd mm yyyy");
    BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
    int day = Integer.parseInt(br.readLine());
    int mon = Integer.parseInt(br.readLine());
    int yr = Integer.parseInt(br.readLine());
    int flag = 0;
    try {
      if (mon <= 0 || mon > 12)
        throw new InvalidDateException();
      else {
        if (mon == 1 || mon == 3 || mon == 5 || mon == 7 || mon == 8 || mon == 10 || mon == 12) {
          if (day >= 1 && day <= 31)
            flag = 1;
          else
            throw new InvalidDateException();
        } else if (mon == 2) {
          if (yr % 4 == 0) {
            if (day >= 1 && day <= 29)
              flag = 1;
            else
              throw new InvalidDateException();
          } else {
            if (day >= 1 && day <= 28)
              flag = 1;
            else
              throw new InvalidDateException();
          }
        } else {
          if (mon == 4 || mon == 6 || mon == 9 || mon == 11) {
            if (day >= 1 && day <= 30)
              flag = 1;
            else
              throw new InvalidDateException();
          }
        }
      }
      if (flag == 1) {
        MyDate dt = new MyDate();
        dt.accept(day, mon, yr);
        dt.display();
      }
    } catch (InvalidDateException mm) {
      System.out.println("Invalid Date");
    }
  }
}
```

Assignment 5

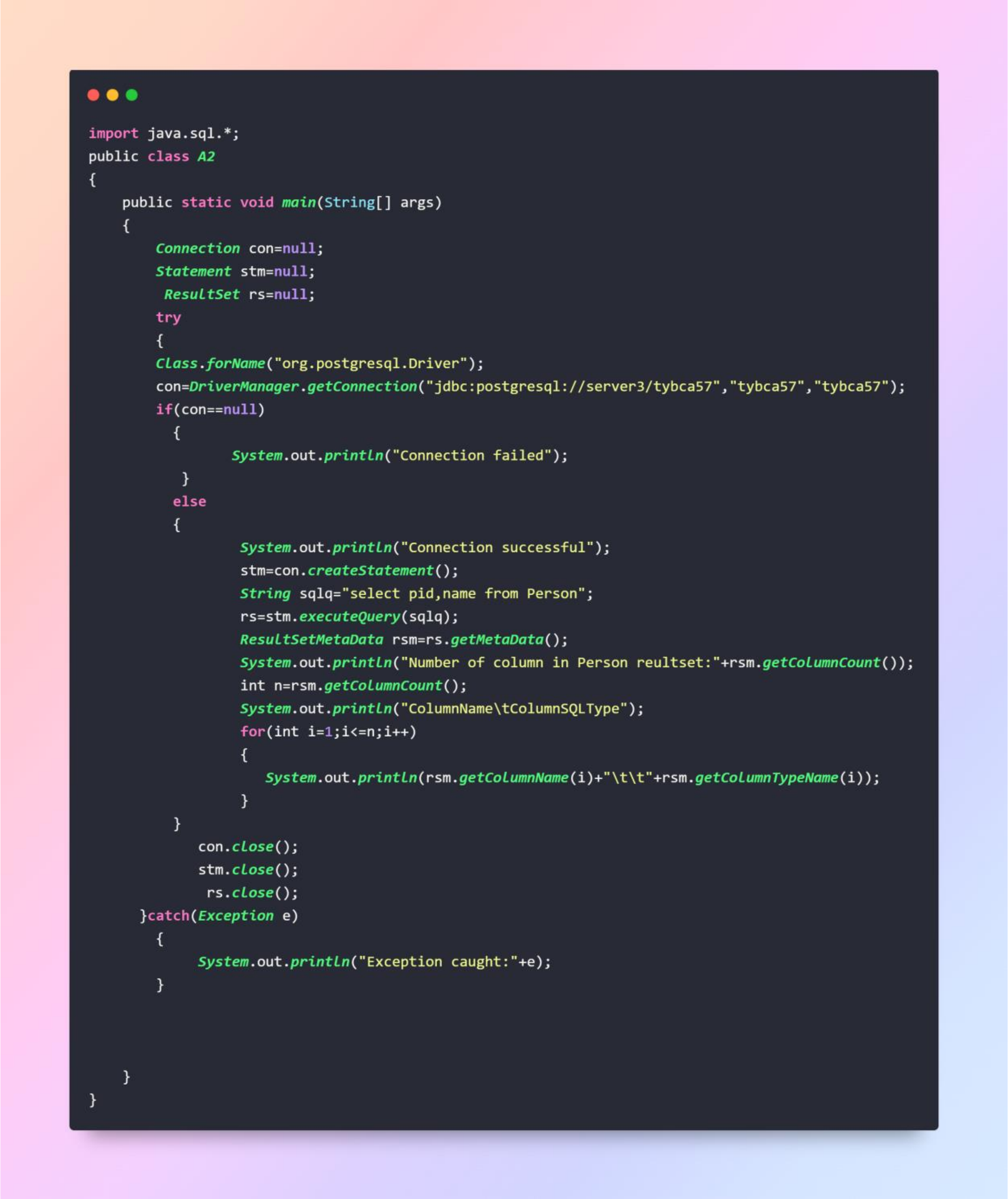
SET A

1. Write a JDBC program to display all the details of the Person table in proper format on the screen. Create a Person table with fields as PID, name, gender, birth_year in PostgreSQL. Insert values in Person table.



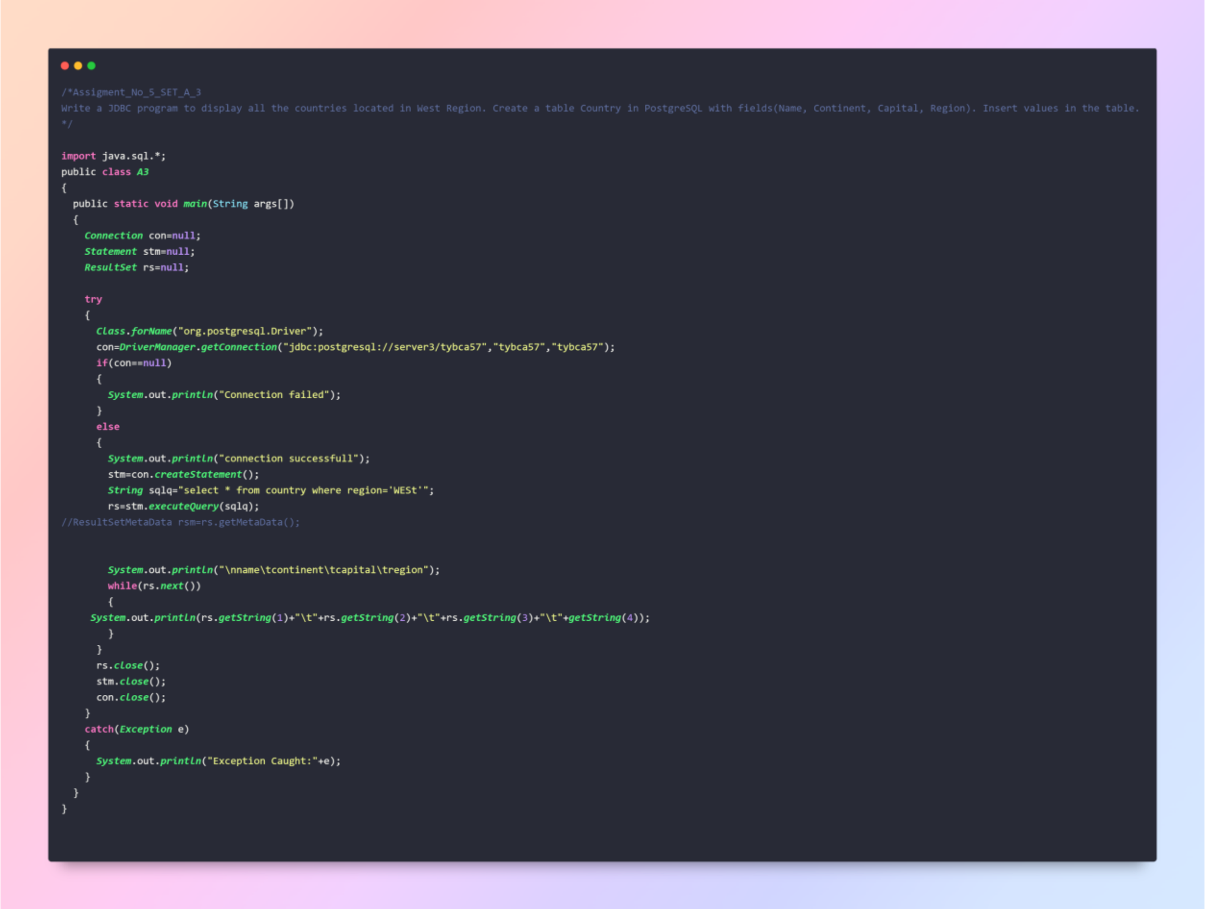
```
import java.sql.*;
public class A1
{
    public static void main(String[] args)
    {
        Connection con=null;
        Statement stm=null;
         ResultSet rs=null;
        try
        {
        Class.forName("org.postgresql.Driver");
        con=DriverManager.getConnection("jdbc:postgresql://server3/tybca57","tybca57","tybca57");
        if(con==null)
          {
                 System.out.println("Connection failed");
           }
          else
          {
                  System.out.println("Connection successful");
                  stm=con.createStatement();
                  String sqlq="select * from Person";
                  rs=stm.executeQuery(sqlq);
                  System.out.println("\nPID\tName\tGender\tBirtyear");
                  while(rs.next())
                  {
                        System.out.println(rs.getInt(1)+"\t"+rs.getString(3)+"\t"+rs.getString(2)+"\t"+    rs.getInt(4));
                  }
          }
             con.close();
             stm.close();
              rs.close();
      }catch(Exception e)
        {
             System.out.println("Exception caught:"+e);
        }
    }
}
```

2. Write a program to display information about the ResultSet like number of columns available in the ResultSet and SQL type of the column. Use Person table. (Use ResultSetMetaData).



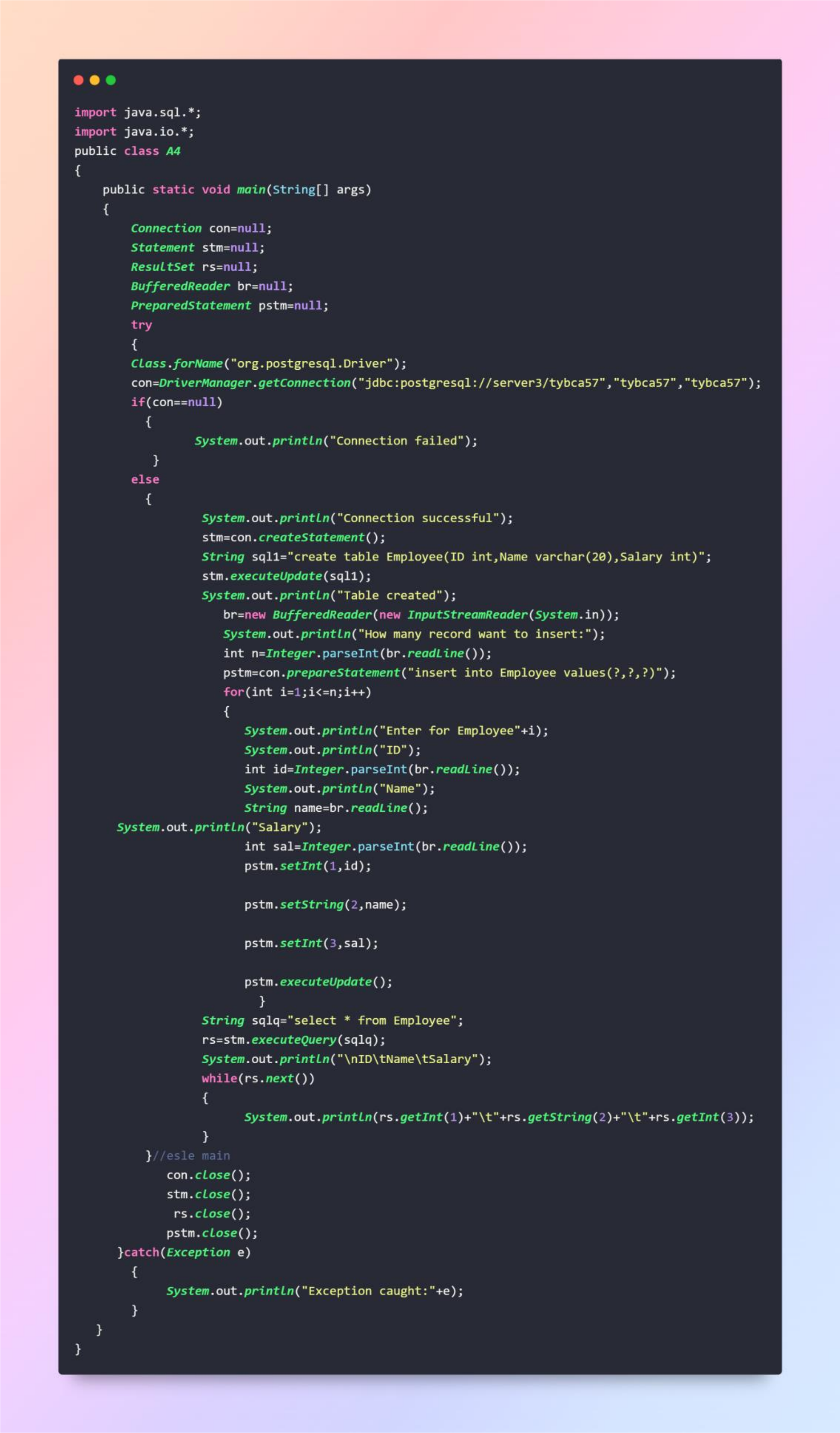
```
import java.sql.*;
public class A2
{
    public static void main(String[] args)
    {
        Connection con=null;
        Statement stm=null;
         ResultSet rs=null;
        try
        {
        Class.forName("org.postgresql.Driver");
        con=DriverManager.getConnection("jdbc:postgresql://server3/tybca57","tybca57","tybca57");
        if(con==null)
          {
                 System.out.println("Connection failed");
           }
          else
          {
                  System.out.println("Connection successful");
                  stm=con.createStatement();
                  String sqlq="select pid,name from Person";
                  rs=stm.executeQuery(sqlq);
                  ResultSetMetaData rsm=rs.getMetaData();
                  System.out.println("Number of column in Person reultset:"+rsm.getColumnCount());
                  int n=rsm.getColumnCount();
                  System.out.println("ColumnName\tColumnSQLType");
                  for(int i=1;i<=n;i++)
                  {
                     System.out.println(rsm.getColumnName(i)+"\t\t"+rsm.getColumnTypeName(i));
                  }
          }
             con.close();
             stm.close();
              rs.close();
      }catch(Exception e)
        {
             System.out.println("Exception caught:"+e);
        }



    }
}
```

3. Write a JDBC program to display all the countries located in West Region. Create a table Country in PostgreSQL with fields (Name, continent, Capital,Region). Insert values in the table.



```
/*Assigment_No_5_SET_A_3
Write a JDBC program to display all the countries located in West Region. Create a table Country in PostgreSQL with fields(Name, Continent, Capital, Region). Insert values in the table.
*/

import java.sql.*;
public class A3
{
  public static void main(String args[])
  {
    Connection con=null;
    Statement stm=null;
    ResultSet rs=null;

    try
    {
      Class.forName("org.postgresql.Driver");
      con=DriverManager.getConnection("jdbc:postgresql://server3/tybca57","tybca57","tybca57");
      if(con==null)
      {
        System.out.println("Connection failed");
      }
      else
      {
        System.out.println("connection successfull");
        stm=con.createStatement();
        String sqlq="select * from country where region='WESt'";
        rs=stm.executeQuery(sqlq);
//ResultSetMetaData rsm=rs.getMetaData();


        System.out.println("\nname\tcontinent\tcapital\tregion");
        while(rs.next())
        {
     System.out.println(rs.getString(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3)+"\t"+getString(4));
        }
      }
      rs.close();
      stm.close();
      con.close();
    }
    catch(Exception e)
    {
      System.out.println("Exception Caught:"+e);
    }
  }
}
```

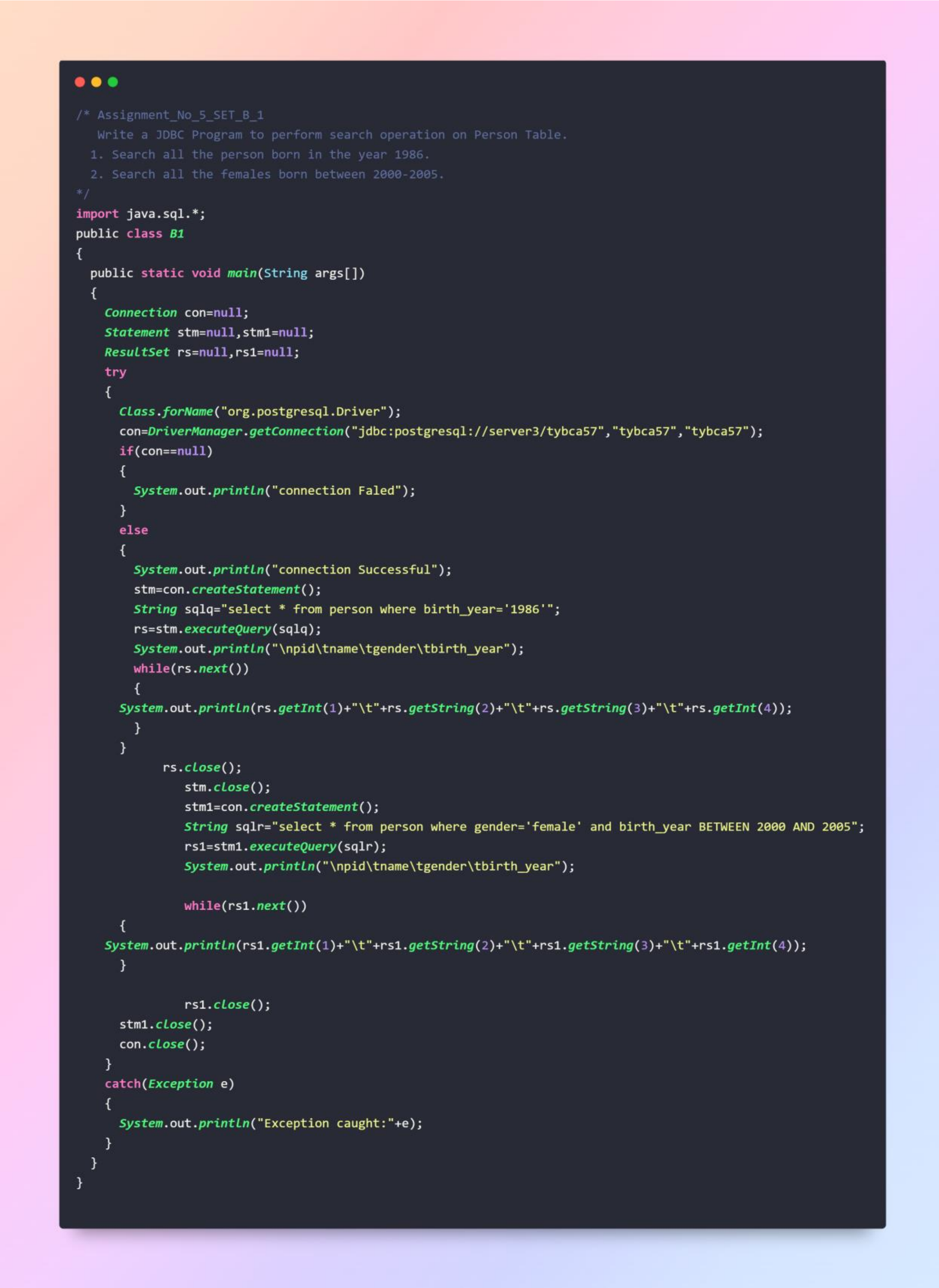
4. Write a JDBC program to insert the records into the table Employee(ID,name,salary) using PreparedStatement interface. Accept details of Employees from user.



```
import java.sql.*;
import java.io.*;
public class A4
{
    public static void main(String[] args)
    {
        Connection con=null;
        Statement stm=null;
        ResultSet rs=null;
        BufferedReader br=null;
        PreparedStatement pstm=null;
        try
        {
        Class.forName("org.postgresql.Driver");
        con=DriverManager.getConnection("jdbc:postgresql://server3/tybca57","tybca57","tybca57");
        if(con==null)
          {
                 System.out.println("Connection failed");
           }
        else
          {
                  System.out.println("Connection successful");
                  stm=con.createStatement();
                  String sql1="create table Employee(ID int,Name varchar(20),Salary int)";
                  stm.executeUpdate(sql1);
                  System.out.println("Table created");
                     br=new BufferedReader(new InputStreamReader(System.in));
                     System.out.println("How many record want to insert:");
                     int n=Integer.parseInt(br.readLine());
                     pstm=con.prepareStatement("insert into Employee values(?,?,?)");
                     for(int i=1;i<=n;i++)
                     {
                        System.out.println("Enter for Employee"+i);
                        System.out.println("ID");
                        int id=Integer.parseInt(br.readLine());
                        System.out.println("Name");
                        String name=br.readLine();
      System.out.println("Salary");
                        int sal=Integer.parseInt(br.readLine());
                        pstm.setInt(1,id);

                        pstm.setString(2,name);

                        pstm.setInt(3,sal);

                        pstm.executeUpdate();
                          }
                  String sqlq="select * from Employee";
                  rs=stm.executeQuery(sqlq);
                  System.out.println("\nID\tName\tSalary");
                  while(rs.next())
                  {
                        System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getInt(3));
                  }
          }//esle main
             con.close();
             stm.close();
              rs.close();
             pstm.close();
      }catch(Exception e)
        {
             System.out.println("Exception caught:"+e);
        }
   }
}
```

SET B

1. Write a JDBC program to perform search operation on Person table. 1. Search all the person born in the year 1986. 2. Search all the females born between 2000- 2005.



```
/* Assignment_No_5_SET_B_1
   Write a JDBC Program to perform search operation on Person Table.
  1. Search all the person born in the year 1986.
  2. Search all the females born between 2000-2005.
*/
import java.sql.*;
public class B1
{
  public static void main(String args[])
  {
    Connection con=null;
    Statement stm=null,stm1=null;
    ResultSet rs=null,rs1=null;
    try
    {
      Class.forName("org.postgresql.Driver");
      con=DriverManager.getConnection("jdbc:postgresql://server3/tybca57","tybca57","tybca57");
      if(con==null)
      {
        System.out.println("connection Faled");
      }
      else
      {
        System.out.println("connection Successful");
        stm=con.createStatement();
        String sqlq="select * from person where birth_year='1986'";
        rs=stm.executeQuery(sqlq);
        System.out.println("\npid\tname\tgender\tbirth_year");
        while(rs.next())
        {
      System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3)+"\t"+rs.getInt(4));
        }
      }
            rs.close();
               stm.close();
               stm1=con.createStatement();
               String sqlr="select * from person where gender='female' and birth_year BETWEEN 2000 AND 2005";
               rs1=stm1.executeQuery(sqlr);
               System.out.println("\npid\tname\tgender\tbirth_year");

               while(rs1.next())
      {
    System.out.println(rs1.getInt(1)+"\t"+rs1.getString(2)+"\t"+rs1.getString(3)+"\t"+rs1.getInt(4));
      }


               rs1.close();
      stm1.close();
      con.close();
    }
    catch(Exception e)
    {
      System.out.println("Exception caught:"+e);
    }
  }
}
```