

Trend.Nxt Core Java L1 Assignment

Topic 1: JVM Concepts and Language Basics

Assignment 1: Write a java program to display "Welcome to Java Programming" and then print your name on a separate line.

Program:

```
package coreJava_assignments;
```

```
public class MainClass {
```

```
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        System.out.println("Welcome to JAVA Programming");  
        System.out.print("N Balaji");  
    }
```

```
}
```

Output:



The screenshot shows an IDE window titled 'MainClass.java' containing the following code:

```
1 package coreJava_assignments;  
2  
3 public class MainClass {  
4  
5     public static void main(String[] args) {  
6         // TODO Auto-generated method stub  
7         System.out.println("Welcome to JAVA Programming");  
8         System.out.print("N Balaji");  
9     }  
10  
11 }  
12
```

Below the code editor, the 'Console' tab is active, displaying the output of the program:

```
<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (19-Apr-2020, 12:51:59 pm)  
Welcome to JAVA Programming  
N Balaji
```

Assignment 2: Write a Java program to print the result of the following operations. Declare variables and initialize them with given values

- a. $-5 + 8 * 6$
- b. $(55+9) \% 9$
- c. $20 + -3*5 / 8$
- d. $5 + 15 / 3 * 2 - 8 \% 3$

Program:

```
package coreJava_assignments;
```

```
public class MainClass {
```

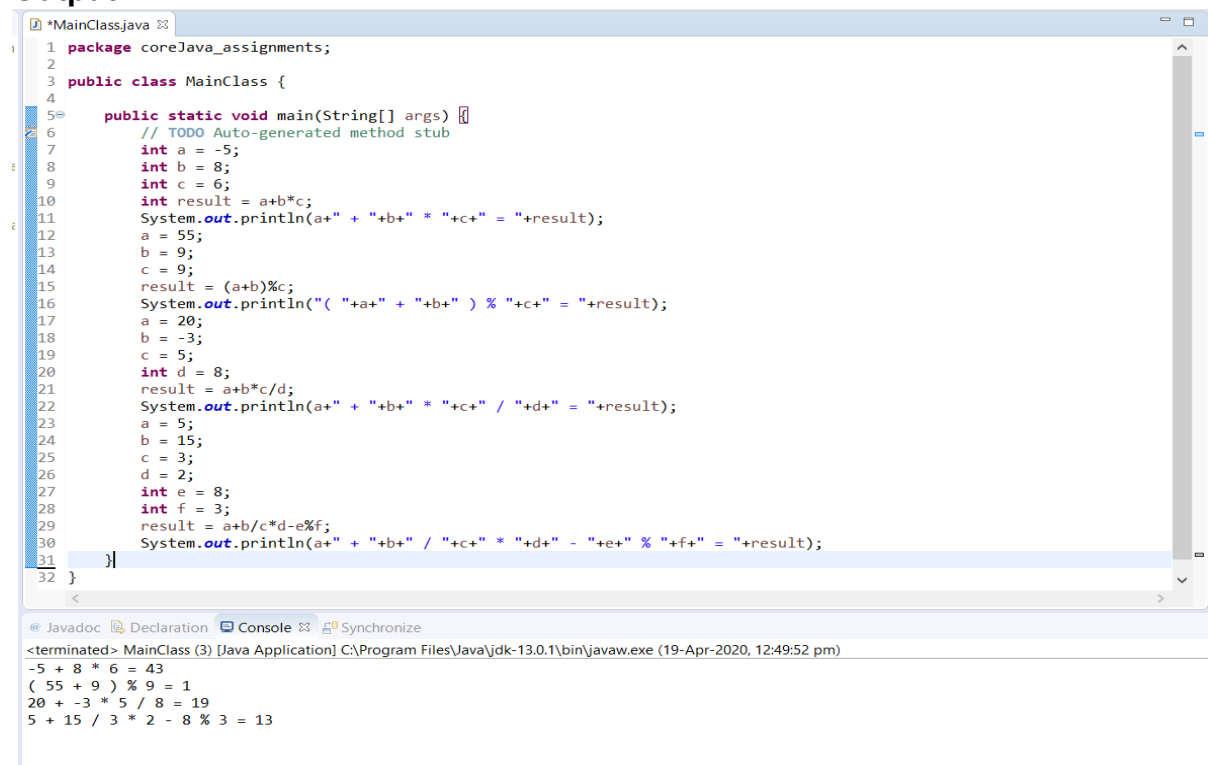
```
    public static void main(String[] args) {
```

```

// TODO Auto-generated method stub
int a = -5;
int b = 8;
int c = 6;
int result = a+b*c;
System.out.println(a+" + "+b+" * "+c+" = "+result);
a = 55;
b = 9;
c = 9;
result = (a+b)%c;
System.out.println("( "+a+" + "+b+" ) % "+c+" = "+result);
a = 20;
b = -3;
c = 5;
int d = 8;
result = a+b*c/d;
System.out.println(a+" + "+b+" * "+c+" / "+d+" = "+result);
a = 5;
b = 15;
c = 3;
d = 2;
int e = 8;
int f = 3;
result = a+b/c*d-e%f;
System.out.println(a+" + "+b+" / "+c+" * "+d+" - "+e+" % "+f+" =
"+result);
}
}

```

Output:



The screenshot shows an IDE window titled '*MainClass.java'. The code is as follows:

```

1 package coreJava_assignments;
2
3 public class MainClass {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int a = -5;
8         int b = 8;
9         int c = 6;
10        int result = a+b*c;
11        System.out.println(a+" + "+b+" * "+c+" = "+result);
12        a = 55;
13        b = 9;
14        c = 9;
15        result = (a+b)%c;
16        System.out.println("( "+a+" + "+b+" ) % "+c+" = "+result);
17        a = 20;
18        b = -3;
19        c = 5;
20        int d = 8;
21        result = a+b*c/d;
22        System.out.println(a+" + "+b+" * "+c+" / "+d+" = "+result);
23        a = 5;
24        b = 15;
25        c = 3;
26        d = 2;
27        int e = 8;
28        int f = 3;
29        result = a+b/c*d-e%f;
30        System.out.println(a+" + "+b+" / "+c+" * "+d+" - "+e+" % "+f+" =
31        "+result);
32    }
33 }

```

The console output at the bottom shows the results of the program:

```

<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (19-Apr-2020, 12:49:52 pm)
-5 + 8 * 6 = 43
( 55 + 9 ) % 9 = 1
20 + -3 * 5 / 8 = 19
5 + 15 / 3 * 2 - 8 % 3 = 13

```

Assignment 3: Write a Java program to convert minutes into a number of years and days.

Program:

```
package coreJava_assignments;
```

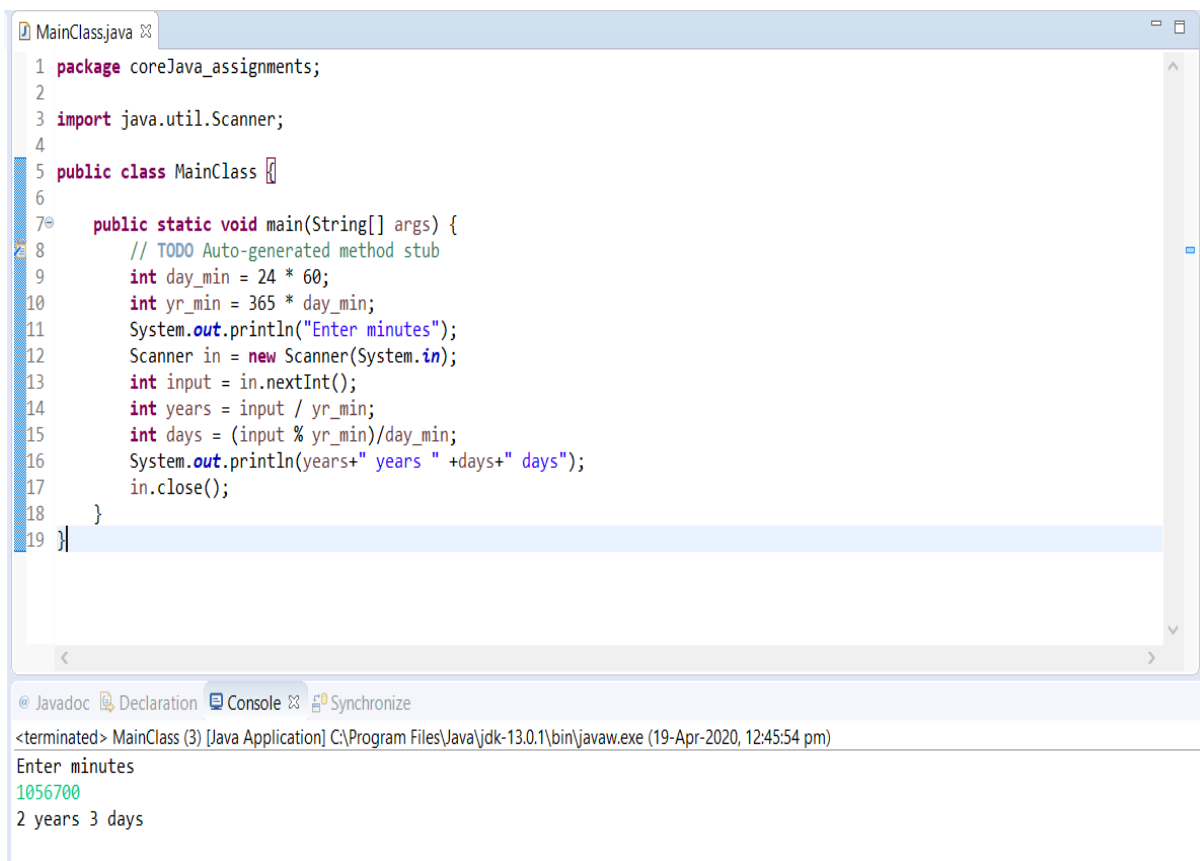
```
import java.util.Scanner;
```

```
public class MainClass {
```

```
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int day_min = 24 * 60;  
        int yr_min = 365 * day_min;  
        System.out.println("Enter minutes");  
        Scanner in = new Scanner(System.in);  
        int input = in.nextInt();  
        int years = input / yr_min;  
        int days = (input % yr_min)/day_min;  
        System.out.println(years+" years " +days+" days");  
        in.close();  
    }
```

```
}
```

Output:



The screenshot shows an IDE window titled 'MainClass.java' containing the Java code from the previous blocks. Below the code editor, the 'Console' tab is active, displaying the program's output. The output shows the prompt 'Enter minutes' followed by the user input '1056700' and the resulting calculation '2 years 3 days'.

```
1 package coreJava_assignments;  
2  
3 import java.util.Scanner;  
4  
5 public class MainClass {  
6  
7     public static void main(String[] args) {  
8         // TODO Auto-generated method stub  
9         int day_min = 24 * 60;  
10        int yr_min = 365 * day_min;  
11        System.out.println("Enter minutes");  
12        Scanner in = new Scanner(System.in);  
13        int input = in.nextInt();  
14        int years = input / yr_min;  
15        int days = (input % yr_min)/day_min;  
16        System.out.println(years+" years " +days+" days");  
17        in.close();  
18    }  
19 }
```

@ Javadoc Declaration Console Synchronize
<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (19-Apr-2020, 12:45:54 pm)
Enter minutes
1056700
2 years 3 days

Assignment 4: Write a program to print month in words, based on input month in numbers. (using switch case)

Program:

```
package coreJava_assignments;

import java.util.Scanner;

public class MainClass {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Enter month number");
        Scanner in = new Scanner(System.in);
        int input = in.nextInt();
        switch(input) {
            case 1 :
                System.out.println("January");
                break;
            case 2 :
                System.out.print("February");
                break;
            case 3 :
                System.out.print("March");
                break;
            case 4 :
                System.out.print("April");
                break;
            case 5 :
                System.out.print("May");
                break;
            case 6 :
                System.out.print("June");
                break;
            case 7 :
                System.out.print("July");
                break;
            case 8 :
                System.out.print("August");
                break;
            case 9 :
                System.out.print("September");
                break;
            case 10 :
                System.out.print("October");
                break;
            case 11 :
                System.out.print("November");
                break;
            case 12 :
```

```

        System.out.print("December");
        break;
    default :
        System.out.print("Invalid Month number");
    }
    in.close();
}
}

```

Output:

```

37     case 9 :
38         System.out.print("September");
39         break;
40     case 10 :
41         System.out.print("October");
42         break;
43     case 11 :

```

<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (19-Apr-2020, 1:01:44 pm)

Enter month number

9

September

Assignment 5: Write a program that will accept a 4-digit number (assume that the user enters only 4-digit nos.) and print the sum of all the 4 digits. For ex: If the number passed is 3629, the program should print “The sum of all the digits entered is 20”.

Program:

```

package coreJava_assignments;

import java.util.Scanner;

public class MainClass {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Enter 4-digit number");
        Scanner in = new Scanner(System.in);
        int input = in.nextInt();
        int input_bkp = input;
        int inpt = input;
        int count = 0, sum = 0, rem;
        while(input_bkp > 0) {
            input_bkp=input_bkp/10;
            count++;
        }
    }
}

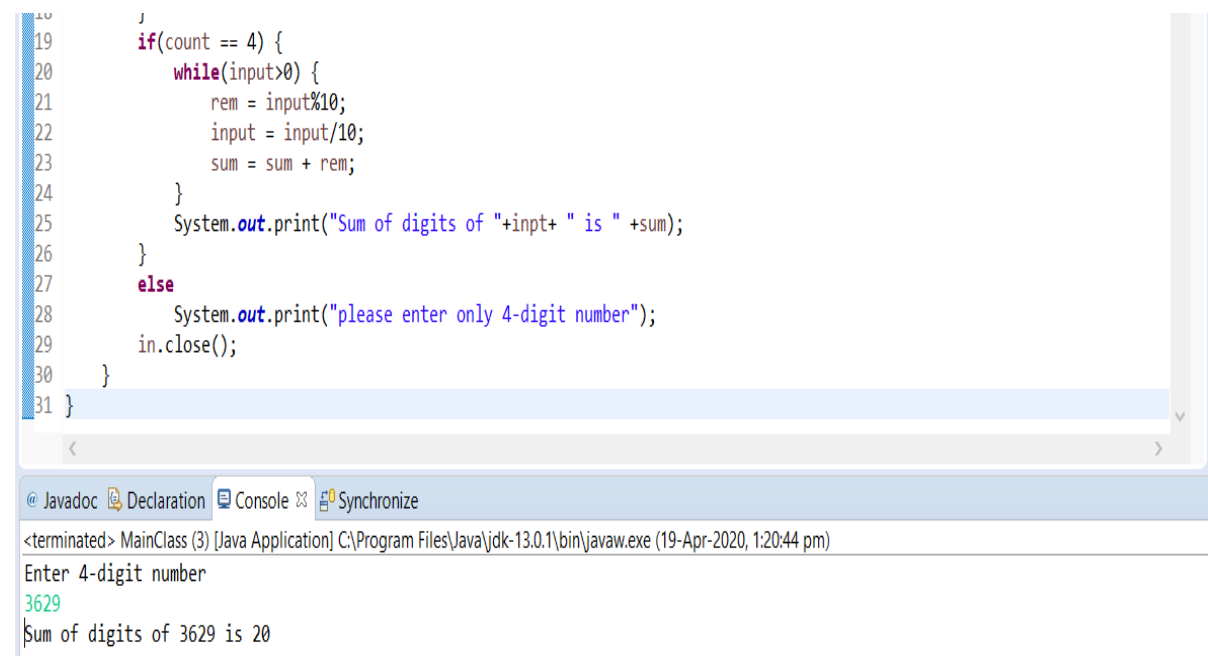
```

```

        if(count == 4) {
            while(input>0) {
                rem = input%10;
                input = input/10;
                sum = sum + rem;
            }
            System.out.print("Sum of digits of "+inpt+ " is " +sum);
        }
        else
            System.out.print("please enter only 4-digit number");
        in.close();
    }
}

```

Output:



The screenshot shows an IDE with a Java file open. The code is the same as the one above. The console window at the bottom shows the execution output:

```

<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (19-Apr-2020, 1:20:44 pm)
Enter 4-digit number
3629
Sum of digits of 3629 is 20

```

Assignment 6: Write a program to find greatest number in an array

Program:

```
package coreJava_assignments;
```

```

public class MainClass {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int[] arr = {18,9,96,19};
        int max = arr[0];
        for(int i = 0;i<arr.length;i++) {
            if(arr[i]>=max)
                max = arr[i];
        }
    }
}

```

```

    }
    System.out.print(max);
}
}

```

Output:

```

MainClass.java
1 package coreJava_assignments;
2
3 public class MainClass {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int[] arr = {18,9,96,19};
8         int max = arr[0];
9         for(int i = 0;i<arr.length;i++) {
10             if(arr[i]>=max)
11                 max = arr[i];
12         }
13         System.out.print(max);
14     }
15 }

```

@ Javadoc Declaration Console Synchronize

<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (19-Apr-2020, 1:33:08 pm)

96

Assignment 7: Write a Java program to calculate the factorial of a number without using any loop.

Program:

```

package coreJava_assignments;

import java.util.Scanner;

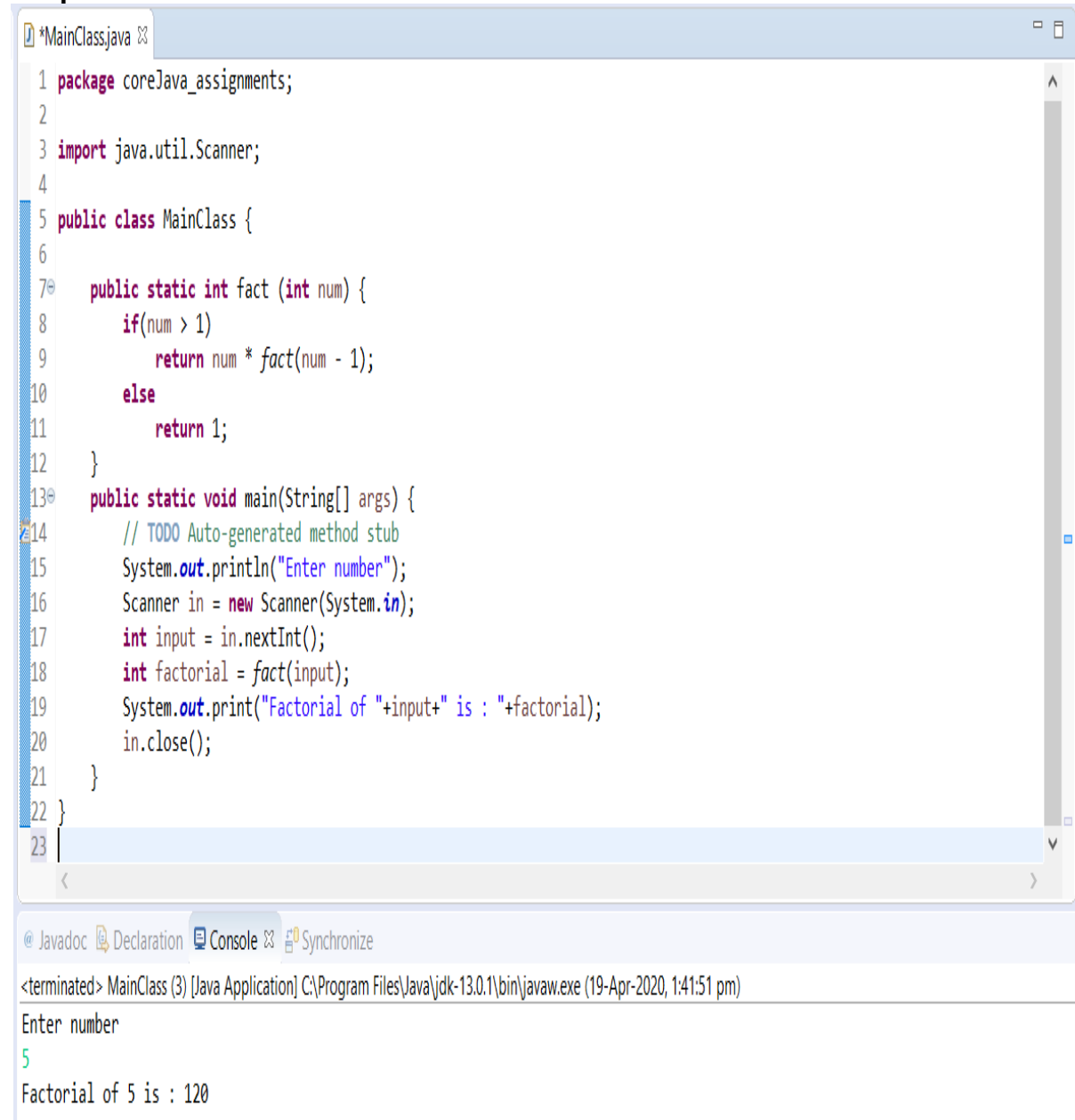
public class MainClass {

    public static int fact (int num) {
        if(num > 1)
            return num * fact(num - 1);
        else
            return 1;
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Enter number");
        Scanner in = new Scanner(System.in);
        int input = in.nextInt();
        int factorial = fact(input);
        System.out.print("Factorial of "+input+" is : "+factorial);
        in.close();
    }
}

```

Output:



```
*MainClass.java
1 package coreJava_assignments;
2
3 import java.util.Scanner;
4
5 public class MainClass {
6
7     public static int fact (int num) {
8         if(num > 1)
9             return num * fact(num - 1);
10        else
11            return 1;
12    }
13    public static void main(String[] args) {
14        // TODO Auto-generated method stub
15        System.out.println("Enter number");
16        Scanner in = new Scanner(System.in);
17        int input = in.nextInt();
18        int factorial = fact(input);
19        System.out.print("Factorial of "+input+" is : "+factorial);
20        in.close();
21    }
22 }
23
```

@ Javadoc Declaration Console Synchronize

<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (19-Apr-2020, 1:41:51 pm)

Enter number

5

Factorial of 5 is : 120

Topic 2: Object Oriented Concepts

Assignment 1: Write a program to create a class Book with the following

- attributes: -isbn, title, author, price

- methods:

i. Initialize the data members through parameterized constructor

ii. displaydetails() to display the details of the book

iii. discountedprice() : pass the discount percent, calculate the discount on price and find the amount to be paid after discount

- task:

Create an object book, initialize the book and display the details along with the discounted price

Program:

Book.java

```
package coreJava_assignments;
```

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

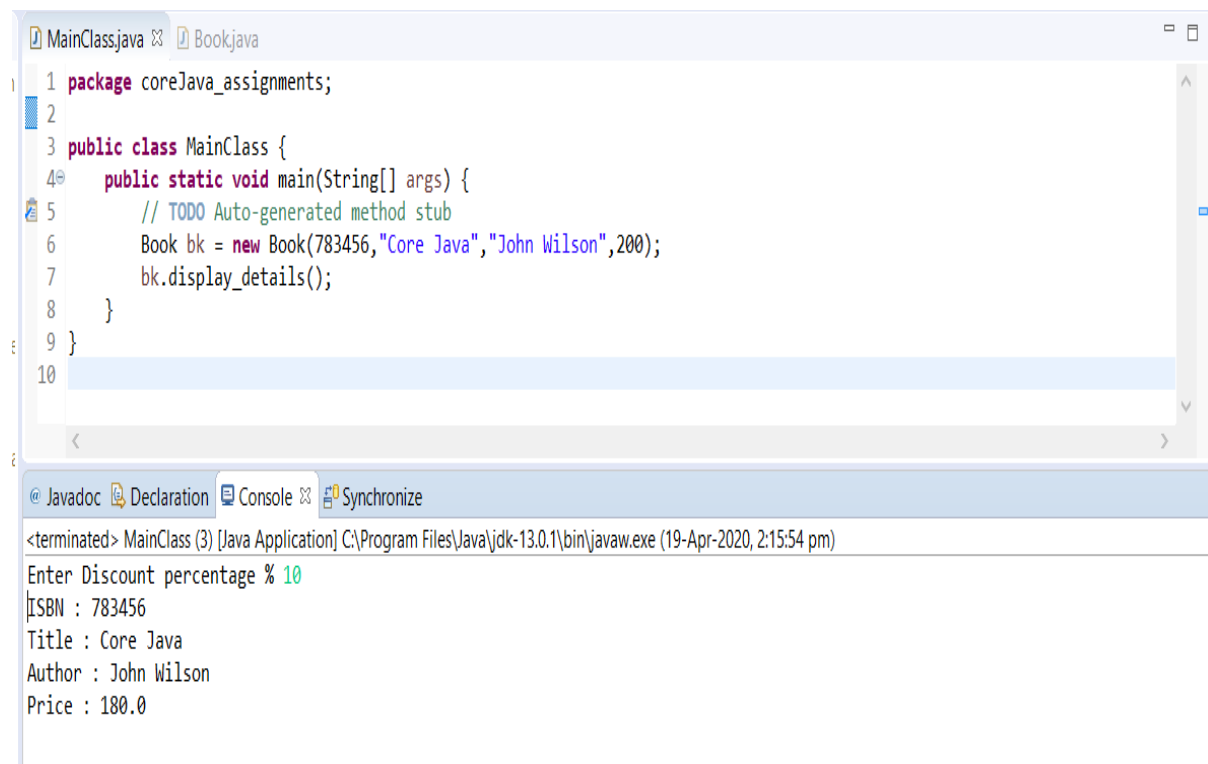
```
public class Book {  
    int isbn;  
    String title,author;  
    float price;  
    /**  
     * @param isbn  
     * @param title  
     * @param author  
     * @param price  
     */  
    public Book(int isbn, String title, String author, float price) {  
        this.isbn = isbn;  
        this.title = title;  
        this.author = author;  
        this.price = price;  
    }  
    public void display_details() {  
        this.price = discountedprice(price);  
        System.out.println("ISBN : " +isbn);  
        System.out.println("Title : " +title);  
        System.out.println("Author : " +author);  
        System.out.println("Price : " +price);  
    }  
    public float discountedprice(float price2) {  
        // TODO Auto-generated method stub  
        Scanner in = new Scanner(System.in);  
        System.out.print("Enter Discount percentage %");  
        float discnt = in.nextFloat();  
        in.close();  
        price2-=price2*(discnt/100);  
        return price2;  
    }  
}
```

MainClass.java

```
package coreJava_assignments;
```

```
public class MainClass {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Book bk = new Book(783456,"Core Java","John Wilson",200);  
        bk.display_details();  
    }  
}
```

Output:



The screenshot shows an IDE with two tabs: MainClass.java and Book.java. The MainClass.java file contains the following code:

```
1 package coreJava_assignments;
2
3 public class MainClass {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         Book bk = new Book(783456, "Core Java", "John Wilson", 200);
7         bk.display_details();
8     }
9 }
10
```

The output window at the bottom shows the execution of the program:

```
<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (19-Apr-2020, 2:15:54 pm)
Enter Discount percentage % 10
ISBN : 783456
Title : Core Java
Author : John Wilson
Price : 180.0
```

Assignment 2:

Define a class named Document that contains a member variable of type String named text that stores any textual content for the document. Create a method named toString that returns the text field and also include a method to set this value. Next, define a class for Email that is derived from Document and includes member variables for the sender, recipient, and title of an email message. Implement appropriate accessor and mutator methods. [An accessor is a member function that accesses the contents of an object but does not modify that object; eg: int getX(return x;) A mutator is a member function that can modify an object void setX(int x){this.x=x;}]The body of the email message should be stored in the inherited variable text. Redefine the toString method to concatenate all text fields

Program:

```
package coreJava_assignments;
```

```
class Document {
    String text;
    public String toString() {
        return text;
    }
    void setString(String str) {
        this.text=str;
    }
}
```

```

class Email extends Document {
    String sender,recipient,title,body;
    public String getSender() {
        return sender;
    }
    public void setSender(String sender) {
        this.sender = sender;
    }
    public String getRecipient() {
        return recipient;
    }
    public void setRecipient(String recipient) {
        this.recipient = recipient;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getBody() {
        return body;
    }
    public void setBody(String body) {
        setString(body);
    }
    public String toString() {
        return "Sender: "+getSender().concat("\nReciever: 
"+getRecipient()).concat("\nTitle: "+getTitle()).concat("\nBody: "+super.toString());
    }
}

public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Email em = new Email();
        em.setSender("balaji@gmail.com");
        em.setRecipient("natarajan@gmail.com");
        em.setTitle("CoreJava L1 assignments");
        em.setBody("Program to implement concepts of inheritance and string
concatenation");
        System.out.print(em.toString());
    }
}

```

Output:

```
44 public class MainClass {
45     public static void main(String[] args) {
46         // TODO Auto-generated method stub
47         Email em = new Email();
48         em.setSender("balaji@gmail.com");
49         em.setRecipient("natarajan@gmail.com");
50         em.setTitle("CoreJava L1 assignments");
51         em.setBody("Program to implement concepts of inheritance and string concatenation");
52         System.out.print(em.toString());
53     }
54 }
```

<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (22-Apr-2020, 10:42:08 am)

Sender: balaji@gmail.com
Reciever: natarajan@gmail.com
Title: CoreJava L1 assignments
Body: Program to implement concepts of inheritance and string concatenation

Assignment 3:

Write a program to create a class Book with the following data members: isbn, title and price. Inherit the class Book to two derived classes: Magazine and Novel with the following data members: Magazine: type Novel: author Populate the details using constructors. Create a magazine and Novel and display the details.

Program:

Book.java

```
package coreJava_assignments;
```

```
public class Book {
    int isbn;
    String title;
    int price;
    /**
     * @param isbn
     * @param title
     * @param price
     */
    public Book(int isbn, String title, int price) {
        this.isbn = isbn;
        this.title = title;
        this.price = price;
    }
}
```

Magazine.java

```
package coreJava_assignments;

public class Magazine extends Book {
    String type;
    public Magazine(int isbn, String title, String type, int price) {
        super(isbn, title, price);
        // TODO Auto-generated constructor stub
        this.type = type;
    }
    public void display() {
        System.out.println("ISBN : " + isbn);
        System.out.println("Title : " + title);
        System.out.println("Type : " + type);
        System.out.println("Price : " + price);
    }
}
```

Novel.java

```
package coreJava_assignments;

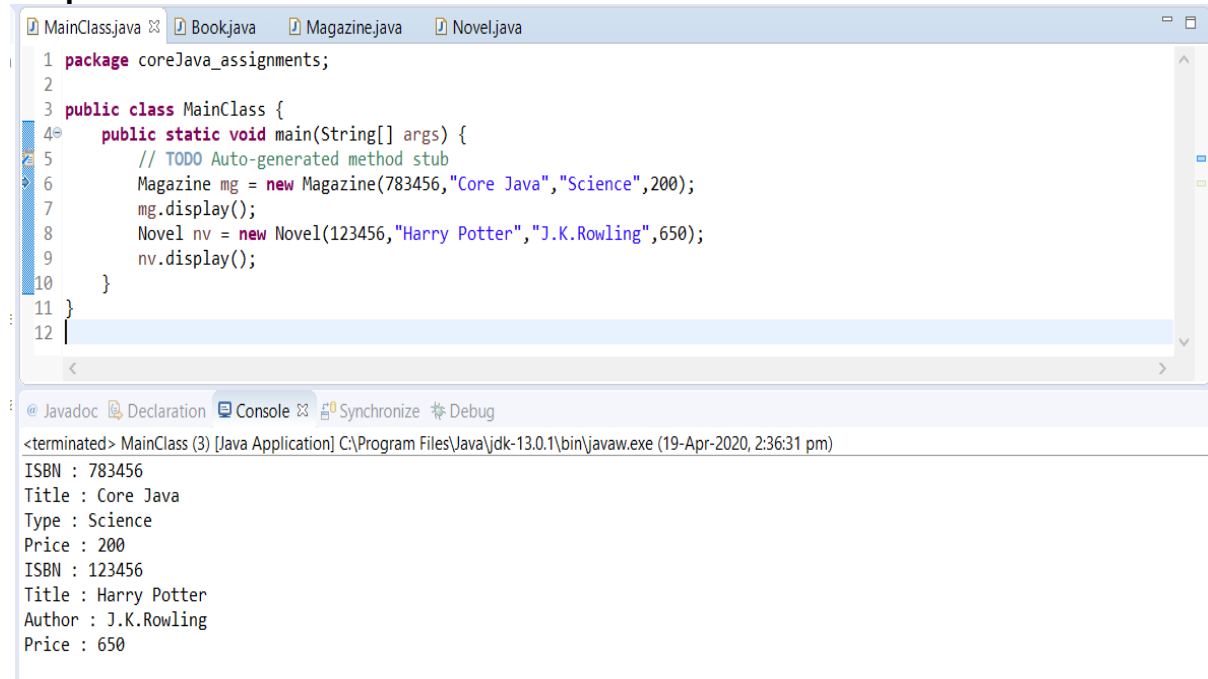
public class Novel extends Book {
    String author;
    public Novel(int isbn, String title, String author, int price) {
        super(isbn, title, price);
        // TODO Auto-generated constructor stub
        this.author = author;
    }
    public void display() {
        System.out.println("ISBN : " + isbn);
        System.out.println("Title : " + title);
        System.out.println("Author : " + author);
        System.out.println("Price : " + price);
    }
}
```

MainClass.java

```
package coreJava_assignments;

public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Magazine mg = new Magazine(783456,"Core Java","Science",200);
        mg.display();
        Novel nv = new Novel(123456,"Harry Potter","J.K.Rowling",650);
        nv.display();
    }
}
```

Output:



The screenshot shows an IDE with a Java project. The main class, MainClass.java, contains the following code:

```
1 package coreJava_assignments;
2
3 public class MainClass {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         Magazine mg = new Magazine(783456,"Core Java","Science",200);
7         mg.display();
8         Novel nv = new Novel(123456,"Harry Potter","J.K.Rowling",650);
9         nv.display();
10    }
11 }
12
```

The console output shows the results of the program execution:

```
<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (19-Apr-2020, 2:36:31 pm)
ISBN : 783456
Title : Core Java
Type : Science
Price : 200
ISBN : 123456
Title : Harry Potter
Author : J.K.Rowling
Price : 650
```

Assignment 4:

Define a class named Payment that contains a member variable of type double that stores the amount of the payment and appropriate accessor and mutator methods. Also create a method named paymentDetails that outputs an English sentence to describe the amount of the payment. Next, define a class named CashPayment that is derived from Payment. This class should redefine the paymentDetails method to indicate that the payment is in cash. Include appropriate constructor(s). Define a class named CreditCardPayment that is derived from Payment. This class should contain member variables for the name on the card, expiration date, and credit card number. Include appropriate constructor(s). Finally, redefine the paymentDetails method to include all credit card information in the printout. Create a main method that creates at least two CashPayment and two CreditCardPayment objects with different values and calls paymentDetails for each.

Program:

```
package coreJava_assignments;

class Payment {
    double amount;
    public double getAmount() {
        return amount;
    }
    void setAmount(double amount) {
        this.amount=amount;
    }
    void paymentDetails() {
        System.out.println("Payment Amount = "+getAmount());
    }
}
```

```

class CashPayment extends Payment {
    public CashPayment(double amount) {
        setAmount(amount);
    }
    void paymentDetails() {
        super.paymentDetails();
        System.out.println("Mode of Payment = Cash");
    }
}

class CreditCardPayment extends Payment {
    String name, expiration_date, credit_card_number;
    public CreditCardPayment(String name, String expiration_date, String
credit_card_number, double amount) {
        this.name = name;
        this.expiration_date = expiration_date;
        this.credit_card_number = credit_card_number;
        setAmount(amount);
    }
    void paymentDetails() {
        super.paymentDetails();
        System.out.println("Mode of Payment = CreditCard");
        System.out.println("Name of the Card: "+name+"\nCredIt Card Number
: "+credit_card_number+"\nExpiration Date : "+expiration_date);
    }
}

public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        CashPayment cp = new CashPayment(10000);
        cp.paymentDetails();
        CreditCardPayment ccp = new CreditCardPayment("Master
Card", "23/04/2022", "2310450012892233", 150000);
        ccp.paymentDetails();
        CashPayment cp2 = new CashPayment(50000);
        cp2.paymentDetails();
        CreditCardPayment ccp2 = new
CreditCardPayment("Visa", "22/04/2022", "2101540021983322", 500000);
        ccp2.paymentDetails();
    }
}

```

Output:

```
38
39 public class MainClass {
40     public static void main(String[] args) {
41         // TODO Auto-generated method stub
42         CashPayment cp = new CashPayment(10000);
43         cp.paymentDetails();
44         CreditCardPayment ccp = new CreditCardPayment("Master Card", "23/04/2022", "2310450012892233", 150000);
45         ccp.paymentDetails();
46         CashPayment cp2 = new CashPayment(50000);
47         cp2.paymentDetails();
48         CreditCardPayment ccp2 = new CreditCardPayment("Visa", "22/04/2022", "2101540021983322", 500000);
49         ccp2.paymentDetails();
50     }
51 }
```

Markers Properties Servers Data Source Explorer Snippets Console

<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (22-Apr-2020, 12:04:06 pm)

```
Payment Amount = 10000.0
Mode of Payment = Cash
Payment Amount = 150000.0
Mode of Payment = CreditCard
Name of the Card: Master Card
Credit Card Number : 2310450012892233
Expiration Date : 23/04/2022
Payment Amount = 50000.0
Mode of Payment = Cash
Payment Amount = 500000.0
Mode of Payment = CreditCard
Name of the Card: Visa
Credit Card Number : 2101540021983322
Expiration Date : 22/04/2022
```

Assignment 5:

Create an abstract class Instrument which is having the abstract function play. Create three more sub classes from Instrument which is Piano, Flute, Guitar. Override the play method inside all three classes printing a message "Piano is playing tan tan tan tan " for Piano class "Flute is playing toot toot toot toot" for Flute class "Guitar is playing tin tin tin " for Guitar class. You must not allow the user to declare an object of Instrument class. Create an array of 10 Instruments. Assign different type of instrument to Instrument reference. Check for the polymorphic behavior of play method. Use the instanceof operator to print that which object stored at which index of instrument array

Program:

```
package coreJava_assignments;
```

```
abstract class Instrument {  
    abstract void play();  
}
```

```
class Piano extends Instrument {  
  
    @Override  
    void play() {  
        // TODO Auto-generated method stub  
        System.out.println("Piano is playing tan tan tan tan ");  
    }  
}
```

```
class Flute extends Instrument {  
    @Override  
    void play() {  
        // TODO Auto-generated method stub  
        System.out.println("Flute is playing toot toot toot toot");  
    }  
}
```

```
class Guitar extends Instrument {  
    @Override  
    void play() {  
        // TODO Auto-generated method stub  
        System.out.println("Guitar is playing tin tin tin ");  
    }  
}
```

```
public class MainClass {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        Instrument imt[] = {new Piano(),new Guitar(),new Flute(),new  
        Flute(),new Guitar(),new Piano(),new Flute(),new Piano(),new Guitar(),new Flute()};  
        for(int i = 0;i<10;i++) {  
            imt[i].play();  
            if(imt[i] instanceof Piano)  
                System.out.println("Piano Instance Stored at "+i);  
            else if(imt[i] instanceof Flute)  
                System.out.println("Flute Instance Stored at "+i);  
            else  
                System.out.println("Guitar Instance Stored at "+i);  
        }  
    }  
}
```

Output:

```
MainClass.java
27 }
28 }
29
30 public class MainClass {
31     public static void main(String[] args) {
32         // TODO Auto-generated method stub
33         Instrument imt[] = {new Piano(),new Guitar(),new Flute(),new Flute(),new Guitar(),new Piano(),new Flute(),new Piano(),new Guitar(),new Flute()};
34         for(int i = 0;i<10;i++) {
35             imt[i].play();
36             if(imt[i] instanceof Piano)
37                 System.out.println("Piano Instance Stored at "+i);
38             else if(imt[i] instanceof Flute)
39                 System.out.println("Flute Instance Stored at "+i);
40             else
41                 System.out.println("Guitar Instance Stored at "+i);
42         }
43     }
44 }
```

<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (22-Apr-2020, 7:12:09 pm)

Piano is playing tan tan tan tan
Piano Instance Stored at 0
Guitar is playing tin tin tin
Guitar Instance Stored at 1
Flute is playing toot toot toot toot
Flute Instance Stored at 2
Flute is playing toot toot toot toot
Flute Instance Stored at 3
Guitar is playing tin tin tin
Guitar Instance Stored at 4
Piano is playing tan tan tan tan
Piano Instance Stored at 5
Flute is playing toot toot toot toot
Flute Instance Stored at 6
Piano is playing tan tan tan tan
Piano Instance Stored at 7
Guitar is playing tin tin tin
Guitar Instance Stored at 8
Flute is playing toot toot toot toot
Flute Instance Stored at 9

Assignment 6:

Write an interface called Playable, with a method void play(); Let this interface be placed in a package called music. Write a class called Veena which implements Playable interface. Let this class be placed in a package music.string Write a class called Saxophone which implements Playable interface. Let this class be placed in a package music.wind

Write another class Test in a package called live. Then,

- Create an instance of Veena and call play() method
- Create an instance of Saxophone and call play() method
- Place the above instances in a variable of type Playable and then call play()

Program:

Playable.java

```
package music;

public interface Playable {
    void play();
}
```

Veena.java

```
package music.string;

import music.Playable;

public class Veena implements Playable {

    public Veena() {
        // TODO Auto-generated constructor stub
    }

    public void play() {
        System.out.println("Veena");
    }
}
```

Saxophone.java

```
package music.wind;

import music.Playable;

public class Saxophone implements Playable {

    public Saxophone() {
        // TODO Auto-generated constructor stub
    }

    public void play() {
        // TODO Auto-generated method stub
        System.out.println("Saxophone");
    }
}
```

Test.java

```
package live;

import music.Playable;
import music.string.Veena;
import music.wind.Saxophone;

public class Test {
```

```

    public Test() {
        // TODO Auto-generated constructor stub
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Veena v = new Veena();
        v.play();
        Saxophone s = new Saxophone();
        s.play();
        Playable pv,ps;
        pv = v;
        ps = s;
        pv.play();
        ps.play();
    }
}

```

Output:

The screenshot shows an IDE with the following components:

- Editor Tabs:** Playable.java, Veena.java, Saxophone.java, and Test.java (active).
- Code Editor:** Displays the source code of Test.java, including package declarations, imports, and the main method.
- Console:** Shows the output of the program, which lists the objects created: Veena, Saxophone, Veena, and Saxophone.

```

1 package live;
2
3 import music.Playable;
4 import music.string.Veena;
5 import music.wind.Saxophone;
6
7 public class Test {
8
9     public Test() {
10         // TODO Auto-generated constructor stub
11     }
12
13     public static void main(String[] args) {
14         // TODO Auto-generated method stub
15         Veena v = new Veena();
16         v.play();
17         Saxophone s = new Saxophone();
18         s.play();
19         Playable pv,ps;
20         pv = v;
21         ps = s;
22         pv.play();
23         ps.play();
24     }
25
26 }
27

```

Console Output:

```

<terminated> Test [Java Application] C:\Program Files\Java\jdk-13.0
Veena
Saxophone
Veena
Saxophone

```

Topic 3: Exceptions, String Concepts

Assignment 1:

Write a program to accept name and age of a person from the command prompt(passed as arguments when you execute the class) and ensure that the age entered is ≥ 18 and < 60 . Display proper error messages. The program must exit gracefully after displaying the error message in case the arguments passed are not proper. (Hint : Create a user defined exception class for handling errors.)

Program:

```
package coreJava_assignments;

import java.util.Scanner;

class MyException extends Exception{
    int a;
    MyException(int b) {
        a=b;
    }
    public String toString() {
        return ("Error "+a+" : Improper age provided");
    }
}

public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner in = new Scanner(System.in);
        String name = in.next();
        int age = in.nextInt();
        in.close();
        try {
            if(age < 18 || age >=60)
                throw new MyException(2);
            else {
                System.out.println("Name : " +name);
                System.out.println("Age : " +age);
            }
        }
        catch(MyException e) {
            System.out.println(e);
        }
    }
}
```

Output:

```
MainClass.java
1 package coreJava_assignments;
2
3 import java.util.Scanner;
4
5 class MyException extends Exception{
6     int a;
7     MyException(int b) {
8         a=b;
9     }
10    public String toString() {
11        return ("Error "+a+" : Improper age provided");
12    }
13 }
14 public class MainClass {
15    public static void main(String[] args) {
16        // TODO Auto-generated method stub
17        Scanner in = new Scanner(System.in);
18        String name = in.next();
19        int age = in.nextInt();
20        in.close();
21        try {
22            if(age < 18 || age >=60)
23                throw new MyException(2);
24            else {
25                System.out.println("Name : " +name);
26                System.out.println("Age : " +age);
27            }
28        }
29        catch(MyException e) {
30            System.out.println(e);
31        }
32    }
33 }
```

@ Javadoc Declaration Console Synchronize Debug

<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-8.0.602\bin\java.exe

Balaji

16

Error 2 : Improper age provided

Assignment 2:

Write a Program to take care of Number Format Exception if user enters values other than integer for calculating average marks of 2 students. The name of the students and marks in 3 subjects are passed as arguments while executing the program.

Program:

```
package coreJava_assignments;
```

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

```
class Student {
    StringBuilder name;
    int m1,m2,m3;
    float avg;
    public Student(StringBuilder name,int m1,int m2,int m3) {
        setName(name);
        setM1(m1);
        setM2(m2);
        setM3(m3);
    }
}
```

```

    void setName(StringBuilder name) {
        this.name = name;
    }
    void setM1(int m1) {
        this.m1 = m1;
    }
    void setM2(int m2) {
        this.m2 = m2;
    }
    void setM3(int m3) {
        this.m3 = m3;
    }
    float calc_avg(int m1,int m2,int m3) {
        return (m1+m2+m3)/3;
    }
    void display() {
        System.out.println("Name of the Student : "+name);
        System.out.println("Mark in subject 1 : "+m1);
        System.out.println("Mark in subject 2 : "+m2);
        System.out.println("Mark in subject 3 : "+m3);
        System.out.println("Average : "+calc_avg(m1,m2,m3));
    }
}

public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner in = new Scanner(System.in);
        int m1,m2,m3;
        StringBuilder name = new StringBuilder();
        for( int i = 0; i<2; i++) {
            try {
                System.out.println("Enter Student" +(i+1)+" Details");
                System.out.println("Enter Name of the Student");
                name.append(in.next());
                System.out.println("Enter Marks obtained by the
student");

                if (in.hasNextInt())
                    m1 = in.nextInt();
                else
                    throw new NumberFormatException();
                if (in.hasNextInt())
                    m2 = in.nextInt();
                else
                    throw new NumberFormatException();
                if (in.hasNextInt())
                    m3 = in.nextInt();
                else
                    throw new NumberFormatException();
                Student st = new Student(name,m1,m2,m3);
                st.display();
            }
        }
    }
}

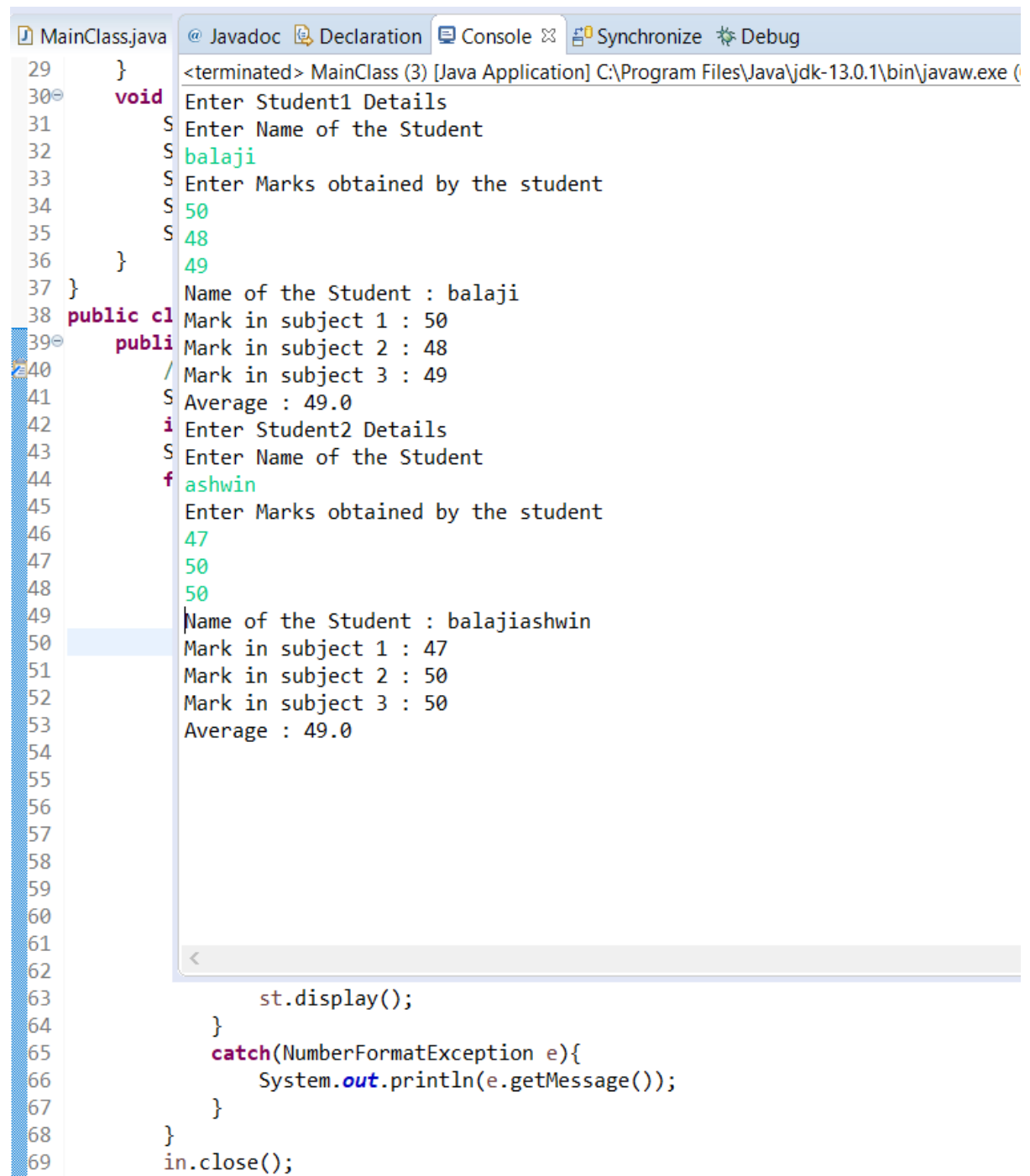
```

```

    }
    catch(NumberFormatException e){
        System.out.println(e.getMessage());
        break;
    }
}
in.close();
}
}

```

Output:

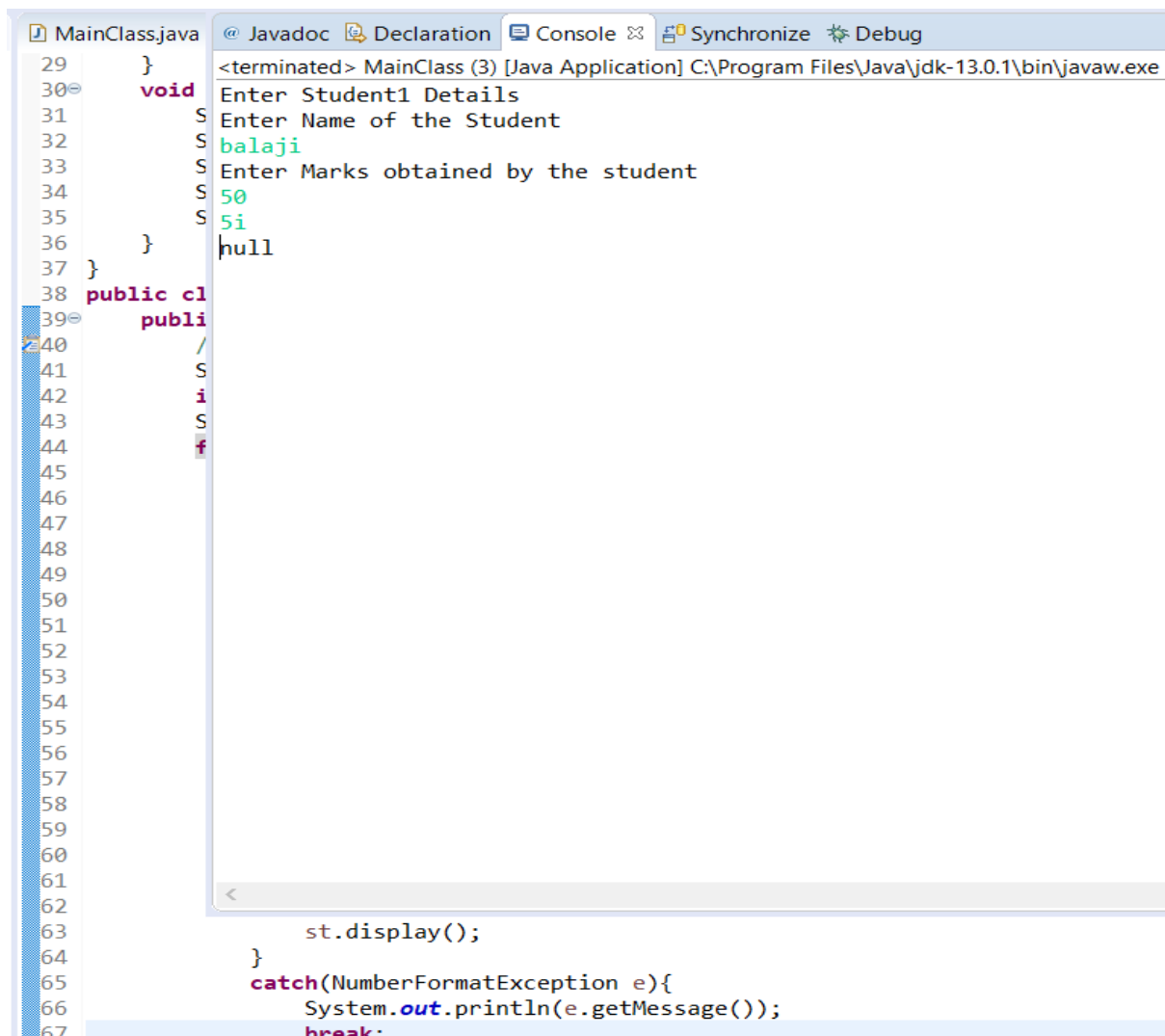


The screenshot shows an IDE with a Java file named 'MainClass.java' and a console window. The code in the editor defines a method 'void' that prompts the user to enter student details. It uses a loop to process input, with a 'break' statement to exit the loop when a 'NumberFormatException' is caught. The console output shows the execution of the program, with user input 'balaji' and '49' for the first student, and 'ashwin' and '50' for the second student. The output also shows the calculated average for each student, which is 49.0 in both cases.

```

MainClass.java  @ Javadoc  Declaration  Console  Synchronize  Debug
29      }
30  void  <terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (
31      S  Enter Student1 Details
32      S  Enter Name of the Student
33      S  balaji
34      S  Enter Marks obtained by the student
35      S  50
36      S  48
37      }  49
38  }  Name of the Student : balaji
39  public cl  Mark in subject 1 : 50
40  publi  Mark in subject 2 : 48
41  /  Mark in subject 3 : 49
42  S  Average : 49.0
43  i  Enter Student2 Details
44  S  Enter Name of the Student
45  f  ashwin
46  Enter Marks obtained by the student
47  47
48  50
49  50
50  Name of the Student : balajiashwin
51  Mark in subject 1 : 47
52  Mark in subject 2 : 50
53  Mark in subject 3 : 50
54  Average : 49.0
55
56
57
58
59
60
61
62
63      st.display();
64      }
65      catch(NumberFormatException e){
66          System.out.println(e.getMessage());
67      }
68  }
69  in.close();

```

```
29     }
30     void Enter Student1 Details
31     S Enter Name of the Student
32     S balaji
33     S Enter Marks obtained by the student
34     S 50
35     S 5i
36     }
37 }
38 public class MainClass {
39     public static void main(String[] args) {
40         // TODO Auto-generated method stub
41         S
42         i
43         S
44         f
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63         st.display();
64     }
65     catch (NumberFormatException e) {
66         System.out.println(e.getMessage());
67         break;
```

Assignment 3:

Write a program to accept 5 integers passed as arguments while executing the class. Find the average of these 5 nos. Use `ArrayIndexOutOfBoundsException` exception to handle situation where the user might have entered less than 5 integers.

Program:

```
package coreJava_assignments;

public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int arr[] = new int[args.length];
        double sum = 0;
        for (int i = 0; i < args.length; i++) {
            arr[i] = Integer.parseInt(args[i]);
            sum += arr[i];
        }
    }
}
```

```

        try {
            if(arr.length!=5)
                throw new ArrayIndexOutOfBoundsException();
            else
                System.out.println("Average of 5 numbers is : "
+(sum/5));
        }
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println(e.getMessage());
            System.exit(0);
        }
    }
}

```

Output:

The screenshot shows the Eclipse IDE with a Java project named 'coreJava_assignments'. The 'MainClass.java' file is open, displaying the following code:

```

1 package coreJava_assignments;
2
3 public class MainClass {
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6         int arr[] = new int[5];
7         double sum = 0;
8         for( int i = 0; i < arr.length; i++) {
9             arr[i] = Integer.parseInt(args[i]);
10            sum += arr[i];
11        }
12        try {
13            if(arr.length!=5)
14                throw new ArrayIndexOutOfBoundsException();
15            else
16                System.out.println("Average of 5 numbers is : "
17                +(sum/5));
18        }
19        catch(ArrayIndexOutOfBoundsException e){
20            System.out.println(e.getMessage());
21            System.exit(0);
22        }
23    }
24 }

```

The 'Console' tab shows the output of the program for three different test cases:

```

C:\Users\Balaji Natarajan\workspace\coreJava_assignments>java MainClass.java 10 20 30 50 40
Average of 5 numbers is : 30.0

C:\Users\Balaji Natarajan\workspace\coreJava_assignments>java MainClass.java 10 20 30 50 40 60
null

C:\Users\Balaji Natarajan\workspace\coreJava_assignments>java MainClass.java 10 20 30
null

```

Assignment 4:

Write a program to check whether the given string is a palindrome or not.

Program:

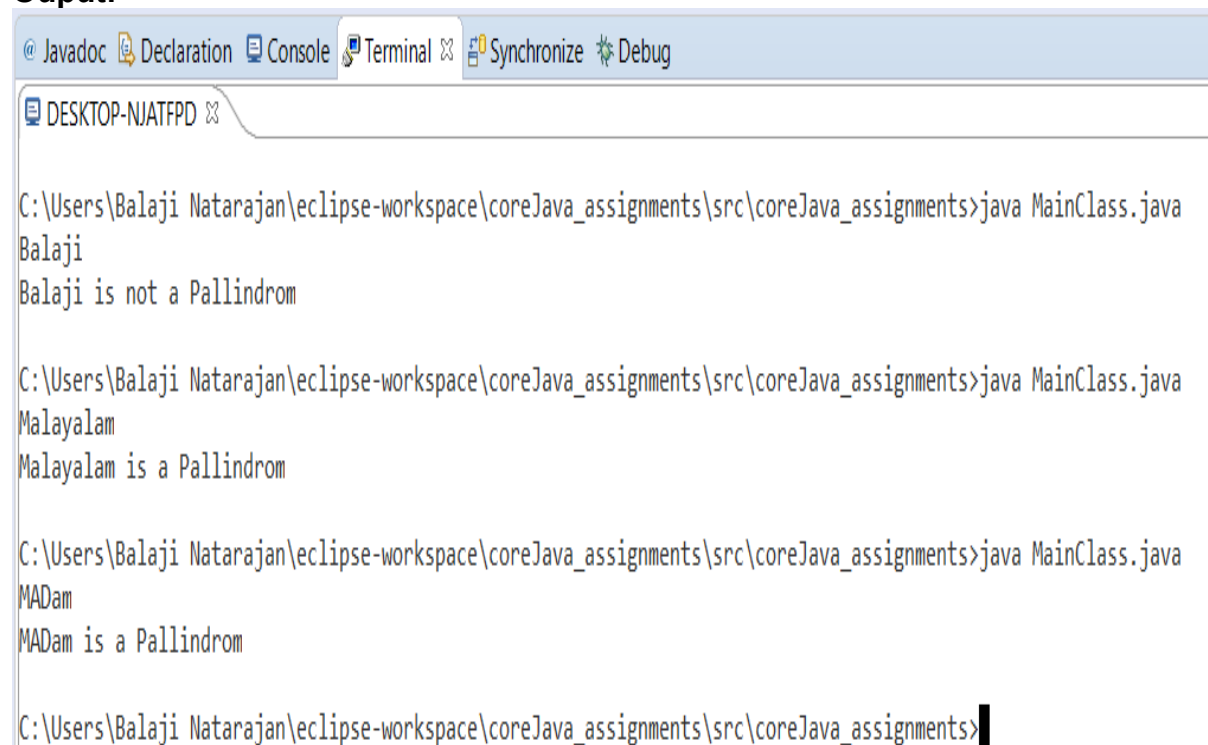
```
package coreJava_assignments;

import java.util.Scanner;

public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner in = new Scanner(System.in);
        String str1 = in.next();
        int l1 = str1.length();
        int flag=0;
        for (int i = 0; i<(l1-1); i++,l1-- ) {

            if(Character.toLowerCase(str1.charAt(i))!=Character.toLowerCase(str1.charAt
(l1-1))))
                flag=1;
        }
        if(flag == 0 )
            System.out.println(str1+" is a Pallindrom");
        else
            System.out.println(str1+" is not a Pallindrom");
        in.close();
    }
}
```

Ouput:



```
@ Javadoc Declaration Console Terminal Synchronize Debug
DESKTOP-NJATFPD
C:\Users\Balaji Natarajan\eclipse-workspace\coreJava_assignments\src\coreJava_assignments>java MainClass.java
Balaji
Balaji is not a Pallindrom

C:\Users\Balaji Natarajan\eclipse-workspace\coreJava_assignments\src\coreJava_assignments>java MainClass.java
Malayalam
Malayalam is a Pallindrom

C:\Users\Balaji Natarajan\eclipse-workspace\coreJava_assignments\src\coreJava_assignments>java MainClass.java
MADam
MADam is a Pallindrom

C:\Users\Balaji Natarajan\eclipse-workspace\coreJava_assignments\src\coreJava_assignments>
```

Assignment 5:

Write a program to check the no. of occurrences of a given character within the given string without using any loop. [Hint: String str="How was your day today"; char c='a'; no. of occurrences of a is=3]

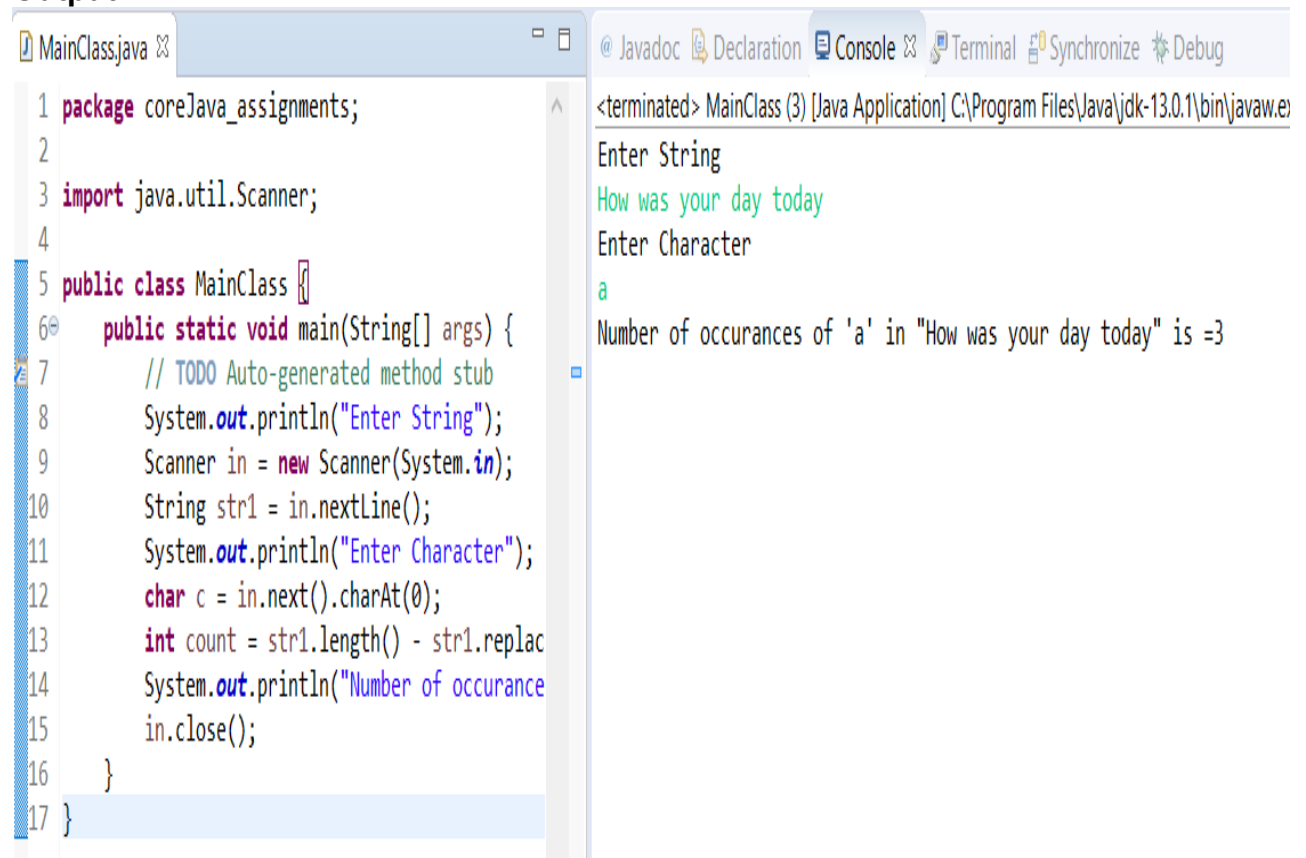
Program:

```
package coreJava_assignments;

import java.util.Scanner;

public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Enter String");
        Scanner in = new Scanner(System.in);
        String str1 = in.nextLine();
        System.out.println("Enter Character");
        char c = in.next().charAt(0);
        int count = str1.length() - str1.replace(Character.toString(c), "").length();
        System.out.println("Number of occurrences of \''+c+'\" in \''+str1+'\" is
        =" + count);
        in.close();
    }
}
```

Output:



The screenshot shows an IDE with two panels. The left panel displays the Java code for MainClass.java, and the right panel shows the console output.

```
1 package coreJava_assignments;
2
3 import java.util.Scanner;
4
5 public class MainClass {
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         System.out.println("Enter String");
9         Scanner in = new Scanner(System.in);
10        String str1 = in.nextLine();
11        System.out.println("Enter Character");
12        char c = in.next().charAt(0);
13        int count = str1.length() - str1.replace
14        System.out.println("Number of occurrence
15        in.close();
16    }
17 }
```

The console output shows the following sequence of events:

```
<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe
Enter String
How was your day today
Enter Character
a
Number of occurrences of 'a' in "How was your day today" is =3
```

Topic 4: Threads, Collection Framework, Garbage Collection

Assignment 1:

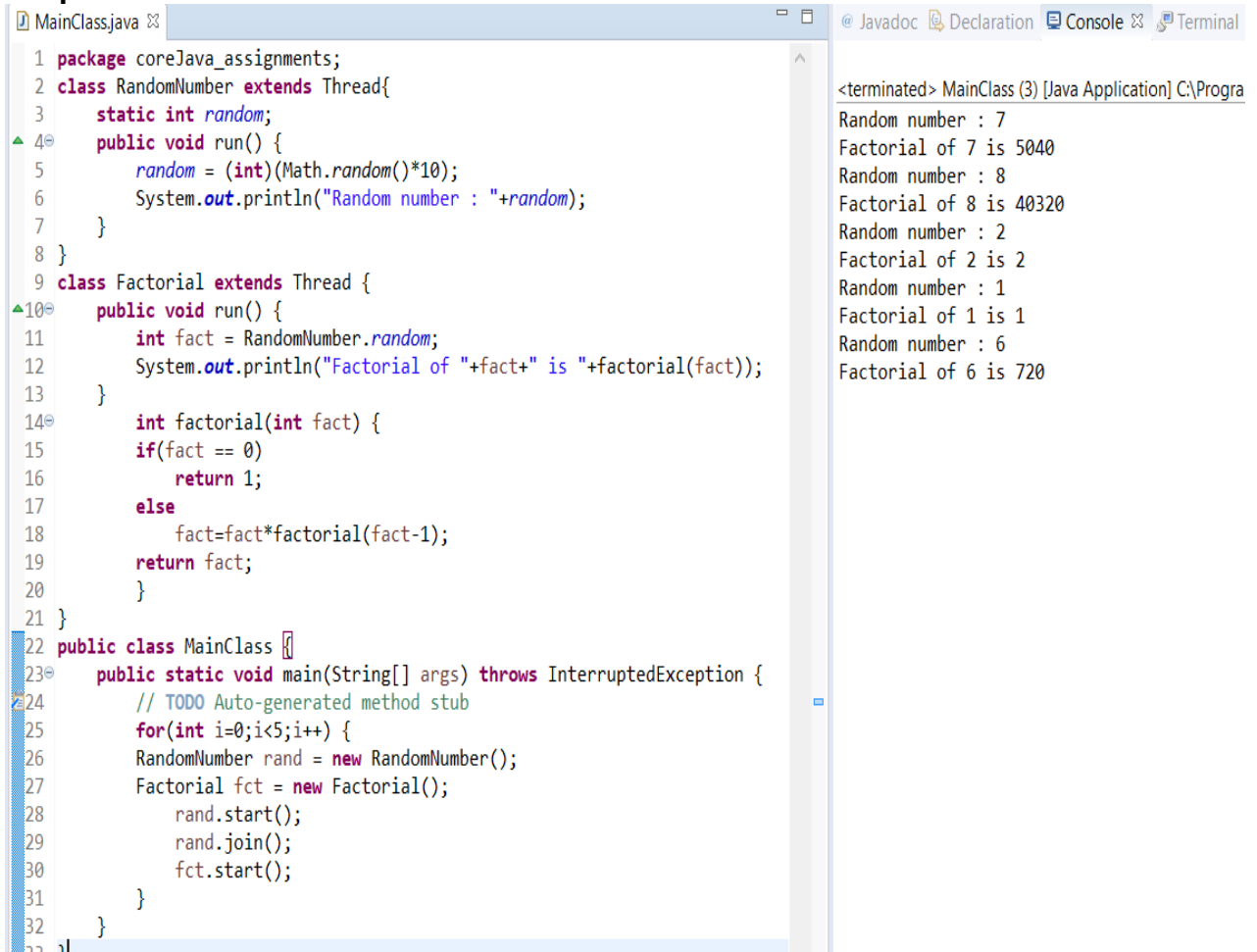
Write a Java Program, where one thread prints a number (Generate a random number using Math.random) and another thread prints the factorial of that given number. Both the outputs should alternate each other. Eg: Number : 2 Factorial of 2 : 2 ; Number : 5 Factorial of 5 : 120 The program can quit after executing 5 times.

Program:

```
package coreJava_assignments;
class RandomNumber extends Thread{
    static int random;
    public void run() {
        random = (int)(Math.random()*10);
        System.out.println("Random number : "+random);
    }
}
class Factorial extends Thread {
    public void run() {
        int fact = RandomNumber.random;
        System.out.println("Factorial of "+fact+" is "+factorial(fact));
    }

    int factorial(int fact) {
        if(fact == 0)
            return 1;
        else
            fact=fact*factorial(fact-1);
        return fact;
    }
}
public class MainClass {
    public static void main(String[] args) throws InterruptedException {
        // TODO Auto-generated method stub
        for(int i=0;i<5;i++) {
            RandomNumber rand = new RandomNumber();
            Factorial fct = new Factorial();
            rand.start();
            rand.join();
            fct.start();
        }
    }
}
```

Output:



The screenshot shows an IDE with a Java file named MainClass.java. The code defines two classes: RandomNumber and Factorial, both extending Thread. RandomNumber has a run() method that prints a random number between 0 and 10. Factorial has a run() method that prints the factorial of a random number. The MainClass has a main() method that creates and starts five instances of RandomNumber and one instance of Factorial. The output on the right shows the execution results: random numbers 7, 8, 2, 1, 6 and their corresponding factorials 5040, 40320, 2, 1, 720.

```
1 package coreJava_assignments;
2 class RandomNumber extends Thread{
3     static int random;
4     public void run() {
5         random = (int)(Math.random()*10);
6         System.out.println("Random number : "+random);
7     }
8 }
9 class Factorial extends Thread {
10    public void run() {
11        int fact = RandomNumber.random;
12        System.out.println("Factorial of "+fact+" is "+factorial(fact));
13    }
14    int factorial(int fact) {
15        if(fact == 0)
16            return 1;
17        else
18            fact=fact*factorial(fact-1);
19        return fact;
20    }
21 }
22 public class MainClass {
23     public static void main(String[] args) throws InterruptedException {
24         // TODO Auto-generated method stub
25         for(int i=0;i<5;i++) {
26             RandomNumber rand = new RandomNumber();
27             Factorial fct = new Factorial();
28             rand.start();
29             rand.join();
30             fct.start();
31         }
32     }
33 }
```

<terminated> MainClass (3) [Java Application] C:\Progra
Random number : 7
Factorial of 7 is 5040
Random number : 8
Factorial of 8 is 40320
Random number : 2
Factorial of 2 is 2
Random number : 1
Factorial of 1 is 1
Random number : 6
Factorial of 6 is 720

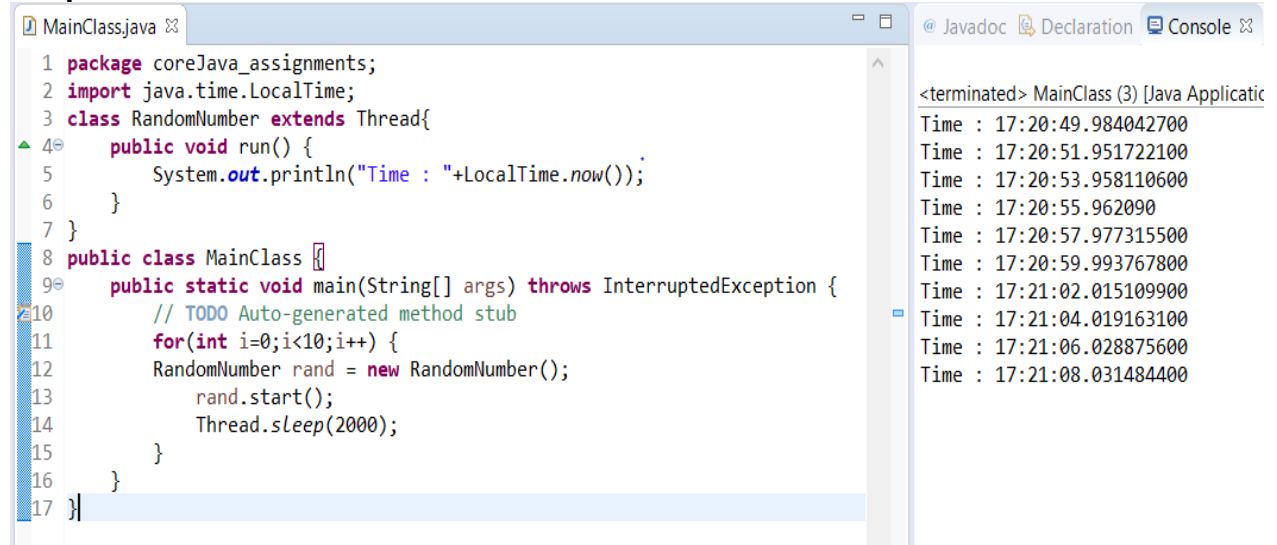
Assignment 2:

Write a Java Program which will print the current time on the console every 2 seconds. After doing this activity for 20 seconds the program quits.

Program:

```
package coreJava_assignments;
import java.time.LocalDateTime;
class RandomNumber extends Thread{
    public void run() {
        System.out.println("Time : "+LocalTime.now());
    }
}
public class MainClass {
    public static void main(String[] args) throws InterruptedException {
        // TODO Auto-generated method stub
        for(int i=0;i<10;i++) {
            RandomNumber rand = new RandomNumber();
            rand.start();
            Thread.sleep(2000);
        }
    }
}
```

Output:



The screenshot shows an IDE with a Java file named `MainClass.java`. The code defines a `RandomNumber` class that extends `Thread` and a `MainClass` with a `main` method. The `main` method creates an instance of `RandomNumber`, starts it, and sleeps for 2000 milliseconds. The `run` method of `RandomNumber` prints the current local time. The console output shows the program running and printing the time at 10-second intervals for 10 iterations.

```
1 package coreJava_assignments;
2 import java.time.LocalDateTime;
3 class RandomNumber extends Thread{
4     public void run() {
5         System.out.println("Time : "+LocalTime.now());
6     }
7 }
8 public class MainClass {
9     public static void main(String[] args) throws InterruptedException {
10        // TODO Auto-generated method stub
11        for(int i=0;i<10;i++) {
12            RandomNumber rand = new RandomNumber();
13            rand.start();
14            Thread.sleep(2000);
15        }
16    }
17 }
```

<terminated> MainClass (3) [Java Applicatic
Time : 17:20:49.984042700
Time : 17:20:51.951722100
Time : 17:20:53.958110600
Time : 17:20:55.962090
Time : 17:20:57.977315500
Time : 17:20:59.993767800
Time : 17:21:02.015109900
Time : 17:21:04.019163100
Time : 17:21:06.028875600
Time : 17:21:08.031484400

Assignment 4:

Write a program creates a HashMap to store name and phone number (Telephone book). When name is give, we can get back the corresponding phone number.

Program:

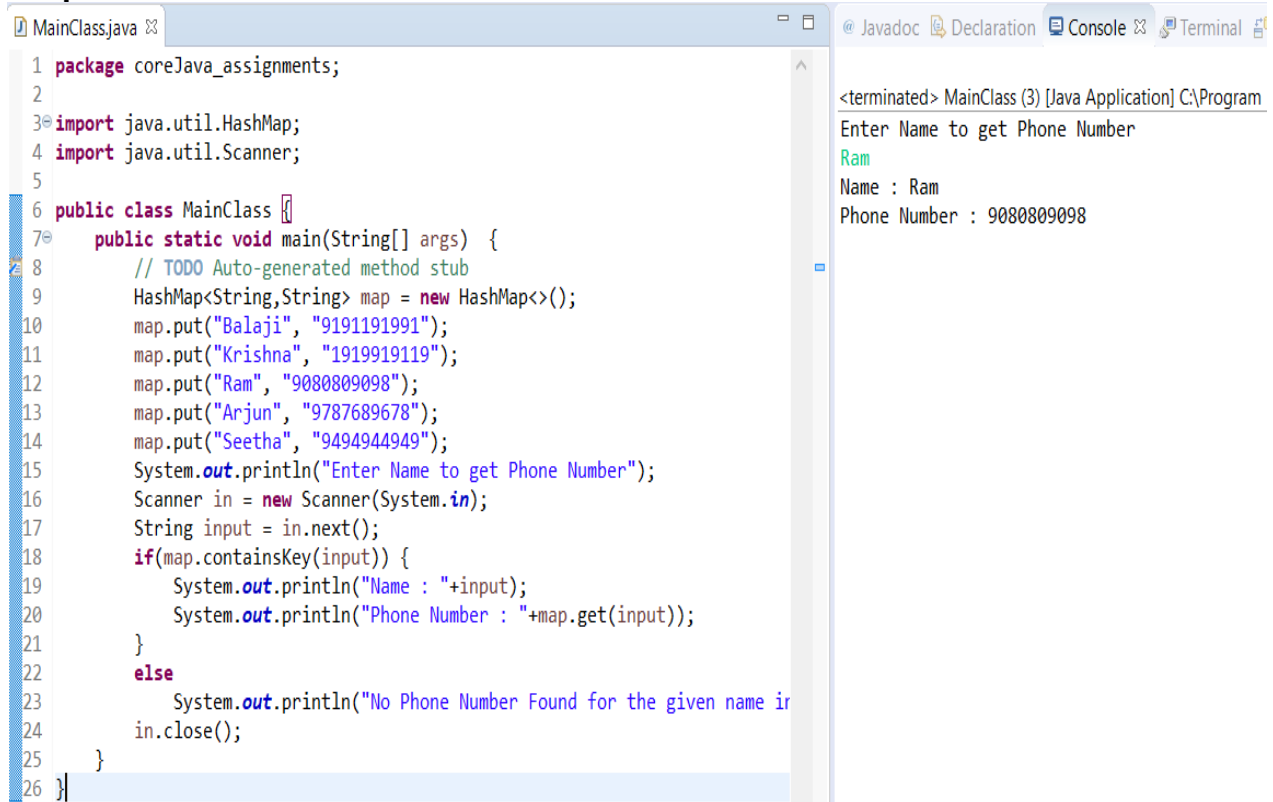
```
package coreJava_assignments;
```

```
import java.util.HashMap;
```

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

```
public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        HashMap<String,String> map = new HashMap<>();
        map.put("Balaji", "9191191991");
        map.put("Krishna", "1919919119");
        map.put("Ram", "9080809098");
        map.put("Arjun", "9787689678");
        map.put("Seetha", "9494944949");
        System.out.println("Enter Name to get Phone Number");
        Scanner in = new Scanner(System.in);
        String input = in.next();
        if(map.containsKey(input)) {
            System.out.println("Name : "+input);
            System.out.println("Phone Number : "+map.get(input));
        }
        else
            System.out.println("No Phone Number Found for the given
name in Phone Book");
        in.close();
    }
}
```

Output:



The screenshot shows an IDE with a Java file named `MainClass.java` and its output in the console. The code defines a `HashMap` with names and phone numbers, prompts the user for a name, and prints the corresponding phone number if found.

```
1 package coreJava_assignments;
2
3 import java.util.HashMap;
4 import java.util.Scanner;
5
6 public class MainClass {
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         HashMap<String,String> map = new HashMap<>();
10        map.put("Balaji", "9191191991");
11        map.put("Krishna", "1919919119");
12        map.put("Ram", "9080809098");
13        map.put("Arjun", "9787689678");
14        map.put("Seetha", "9494944949");
15        System.out.println("Enter Name to get Phone Number");
16        Scanner in = new Scanner(System.in);
17        String input = in.next();
18        if(map.containsKey(input)) {
19            System.out.println("Name : "+input);
20            System.out.println("Phone Number : "+map.get(input));
21        }
22        else
23            System.out.println("No Phone Number Found for the given name in");
24        in.close();
25    }
26 }
```

The console output shows the program execution: `<terminated> MainClass (3) [Java Application] C:\Program`, followed by the prompt `Enter Name to get Phone Number`, the input `Ram`, and the output `Name : Ram` and `Phone Number : 9080809098`.

Assignment 5:

Write a program to store a group of employee names into a `HashSet`, retrieve the elements one by one using an `Iterator`.

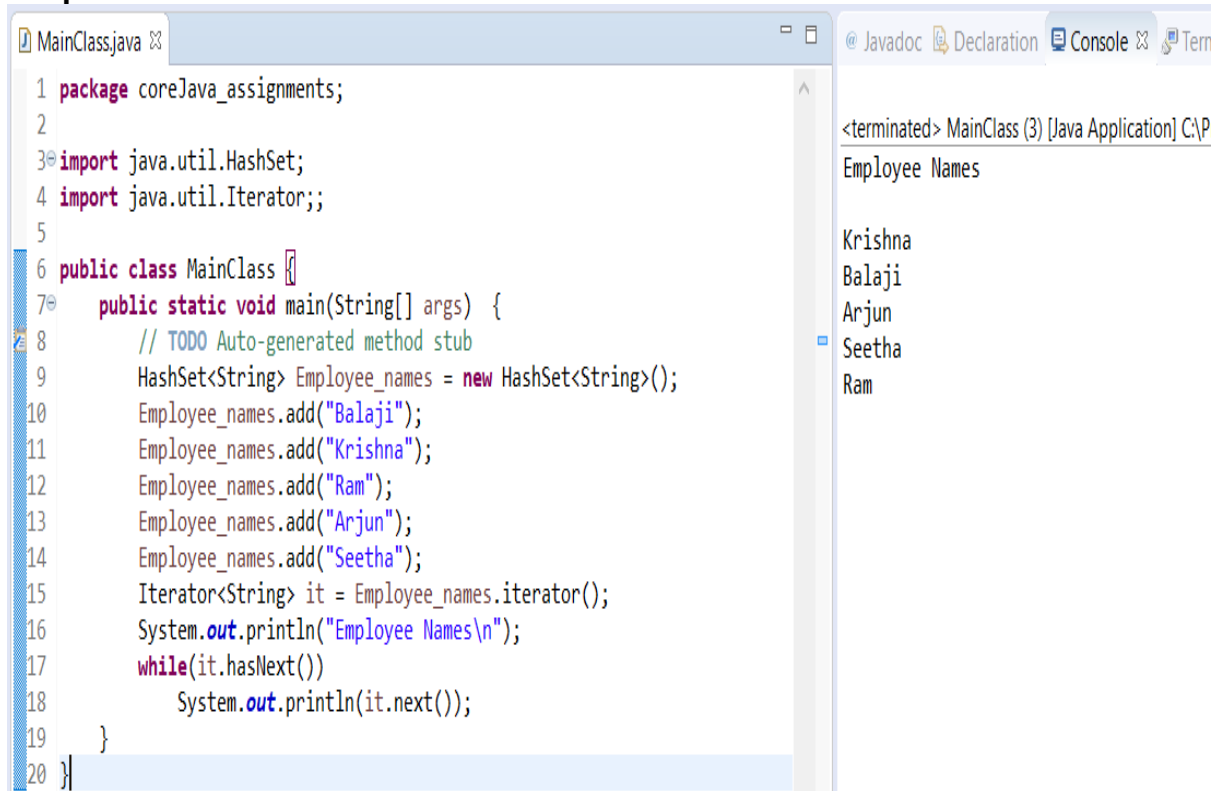
Program:

```
package coreJava_assignments;

import java.util.HashSet;
import java.util.Iterator;;

public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        HashSet<String> Employee_names = new HashSet<String>();
        Employee_names.add("Balaji");
        Employee_names.add("Krishna");
        Employee_names.add("Ram");
        Employee_names.add("Arjun");
        Employee_names.add("Seetha");
        Iterator<String> it = Employee_names.iterator();
        System.out.println("Employee Names\n");
        while(it.hasNext())
            System.out.println(it.next());
    }
}
```


Output:



The screenshot shows an IDE with two panels. The left panel displays the source code for `MainClass.java`. The code defines a package `coreJava_assignments`, imports `java.util.HashSet` and `java.util.Iterator`, and defines a `MainClass` with a `main` method. The `main` method creates a `HashSet` of employee names, adds five names, and prints them using an `Iterator`. The right panel shows the console output, which displays the employee names: Krishna, Balaji, Arjun, Seetha, and Ram.

```
1 package coreJava_assignments;
2
3 import java.util.HashSet;
4 import java.util.Iterator;;
5
6 public class MainClass {
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         HashSet<String> Employee_names = new HashSet<String>();
10        Employee_names.add("Balaji");
11        Employee_names.add("Krishna");
12        Employee_names.add("Ram");
13        Employee_names.add("Arjun");
14        Employee_names.add("Seetha");
15        Iterator<String> it = Employee_names.iterator();
16        System.out.println("Employee Names\n");
17        while(it.hasNext())
18            System.out.println(it.next());
19    }
20 }
```

<terminated> MainClass (3) [Java Application] C:\P
Employee Names

Krishna
Balaji
Arjun
Seetha
Ram

Assignment 6:

Develop a java class that has finalize method which displays “Finalize method called”. Create another class which creates objects of the previous class and it uses the same object reference for creating these objects. For example, if A1 is the class name, then the objects are created as below :

```
A1 a = new A1();
a = new A1();
a = new A1();
```

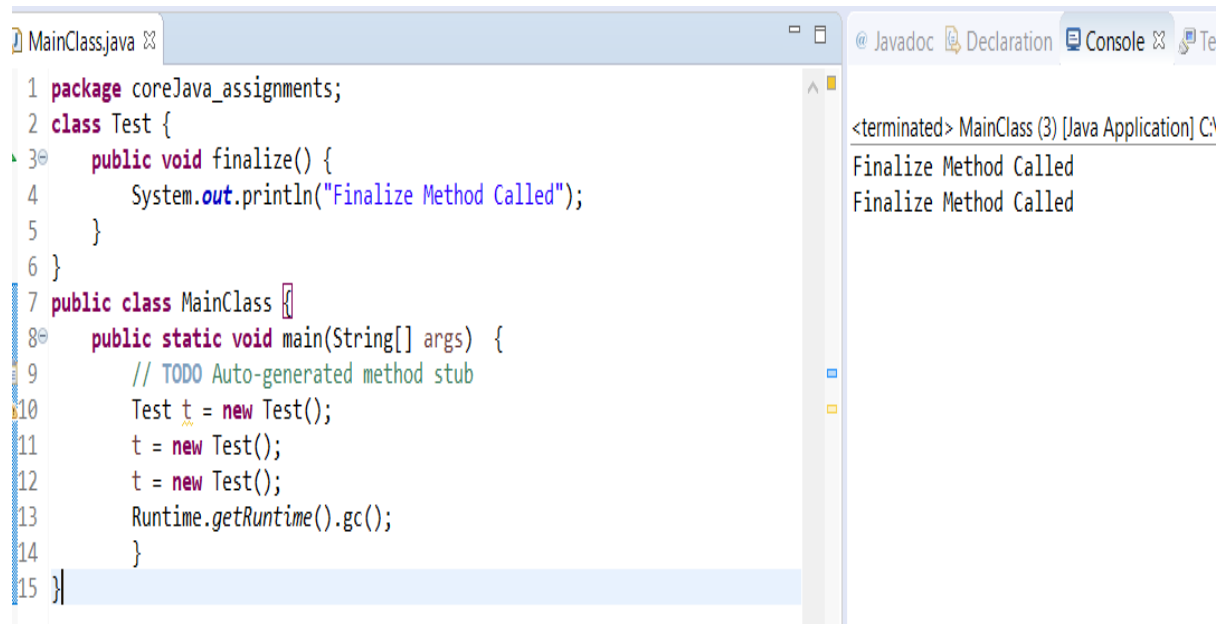
When the statement `Runtime.getRuntime().gc()` is invoked, how many times the finalize method is called

Program:

```
package coreJava_assignments;
class Test {
    public void finalize() {
        System.out.println("Finalize Method Called");
    }
}
public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Test t = new Test();
        t = new Test();
        t = new Test();
        Runtime.getRuntime().gc();
    }
}
```

Output:

Finalize method is called 2 times.



The screenshot shows an IDE with a file named `MainClass.java`. The code defines a `Test` class with a `finalize()` method that prints "Finalize Method Called". The `MainClass` has a `main` method that creates three instances of `Test` and calls `Runtime.getRuntime().gc();`. The console output shows the message "<terminated> MainClass (3) [Java Application] C:\\" followed by two occurrences of "Finalize Method Called".

```
1 package coreJava_assignments;
2 class Test {
3     public void finalize() {
4         System.out.println("Finalize Method Called");
5     }
6 }
7 public class MainClass {
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10        Test t = new Test();
11        t = new Test();
12        t = new Test();
13        Runtime.getRuntime().gc();
14    }
15 }
```

<terminated> MainClass (3) [Java Application] C:\
Finalize Method Called
Finalize Method Called

Topic 5: Command Line Args, System Properties, Packaging

Assignment 1:

Create a package called test package; Define a class called foundation inside the test package; Inside the class, you need to define 4 integer variables; Var1 as private; Var2 as default; Var3 as protected; Var4 as public; Import this class and packages in another class. Try to access all 4 variables of the foundation class and see what variables are accessible and what are not accessible.

Program:

Foundation.java

```
package test;

public class Foundation {
    private String a = "Private variable";
    String b = "Default Variable";
    protected String c = "Protected variable";
    public String d = "Public variable";
}
```

Basic.java

```
package test;

public class Basic {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Foundation F = new Foundation();
        //System.out.println(F.a);
    }
}
```

```

        System.out.println(F.b);
        System.out.println(F.c);
        System.out.println(F.d);
    }
}

```

Output:

Except Private variable all other variables are visible outside the class within the package.

```

1 package test;
2
3 public class Basic {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Foundation F = new Foundation();
8         System.out.println(F.a);
9         System.out.println(F.b);
10        System.out.println(F.c);
11        System.out.println(F.d);
12    }
13
14 }

```

Console Output:

```

<terminated> Basic [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (05-Jun-2020, 8:1
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    The field Foundation.a is not visible
    at test.Basic.main(Basic.java:8)

```

```

1 package test;
2
3 public class Basic {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Foundation F = new Foundation();
8         //System.out.println(F.a);
9         System.out.println(F.b);
10        System.out.println(F.c);
11        System.out.println(F.d);
12    }
13
14 }

```

Console Output:

```

<terminated> Basic [Java Application] C:\Program Fil
Default Variable
Protected variable
Public variable

```

Assignment 2:

Write a Program to accept two Strings Wipro Bangalore as command line arguments and print the output “Wipro Technologies Bangalore” If the command line is “ABC Mumbai”, then it should print “ABC Technologies Mumbai” .

Program:

```
package coreJava_assignments;
```

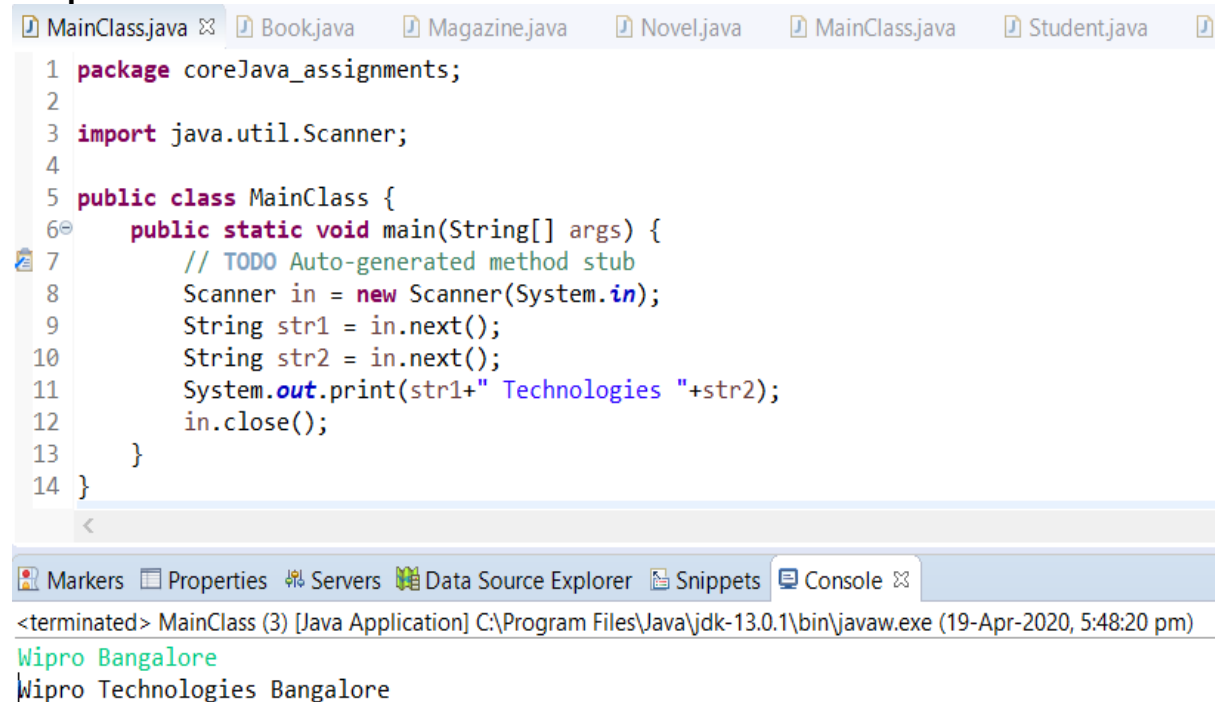
```
import java.util.Scanner;
```

```

public class MainClass {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner in = new Scanner(System.in);
        String str1 = in.next();
        String str2 = in.next();
        System.out.print(str1+" Technologies "+str2);
        in.close();
    }
}

```

Output:



The screenshot shows an IDE with a project named 'coreJava_assignments'. The 'MainClass.java' file is open, displaying the following code:

```

1 package coreJava_assignments;
2
3 import java.util.Scanner;
4
5 public class MainClass {
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         Scanner in = new Scanner(System.in);
9         String str1 = in.next();
10        String str2 = in.next();
11        System.out.print(str1+" Technologies "+str2);
12        in.close();
13    }
14 }

```

The IDE's interface includes tabs for 'MainClass.java', 'Book.java', 'Magazine.java', 'Novel.java', 'MainClass.java', and 'Student.java'. The bottom panel shows the 'Console' tab with the following output:

```

<terminated> MainClass (3) [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (19-Apr-2020, 5:48:20 pm)
Wipro Bangalore
Wipro Technologies Bangalore

```