Yanuar Thaif Chalil Candra

21241720004 / Ti – 1i / 29

Experiment 1

Experiment1.java > Experiment1

```java
public class Experiment1
{
    static int factorialRecursive(int n)
    {
        if (n == 0)
        {
            return (1);
        }
        else
        {
            return (n * factorialRecursive(n - 1));
        }
    }

    static int factorialIterative(int n)
    {
        int factor = 1;
        for (int i = n; i >= 1; i--)
        {
            factor = factor * i;
        }
        return factor;
    }

    Run | Debug
    public static void main(String[] args)
    {
        System.out.println(factorialRecursive(n: 5));
        System.out.println(factorialIterative(n: 5));
    }
}
```

```
D:\hiya kuliah\P_Daspro\java>javac Experiment1.java && java Experiment1
120
120
```

Experiment 2

Experiment2.java > Experiment2

```java
public class Experiment2
{
    static int calculatePower(int x, int y)
    {
        if (y == 0)
        {
            return (1);
        }
        else
        {
            return (x * calculatePower(x, y -1));
        }
    }

    Run | Debug
    public static void main(String[] args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.print(s: "Enter a number: ");
        int number = sc.nextInt();
        System.out.print(s: "Enter the exponent: ");
        int exponent = sc.nextInt();
        System.out.println(calculatePower(number, exponent));
    }
}
```

```
PS D:\hiya kuliah\P_Daspro\java>  d:; cd 'd:\hiya kuliah\P_Das
dt_socket,server=n,suspend=y,address=localhost:59575' '-cp' '(
0cfaa701\redhat.java\jdt_ws\java_188f3e68\bin' 'Experiment2'
Enter a number: 5
Enter the exponent: 4
625
```

Experiment 3

```java
import java.util.Scanner;
public class Experiment3
{
    static double calculateInterest(double balance, int year)
    {
        if (year == 0)
        {
            return (balance);
        }
        else
        {
            return (1.11 * calculateInterest(balance, year - 1));
        }
    }

    public static void main(String[] args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.print("Enter the opening balance: ");
        double openingBalance = sc.nextDouble();
        System.out.print("Enter the duration of saving (years): ");
        int year = sc.nextInt();
        System.out.print("Amount of money after " + year + " years: ");
        System.out.println((int) calculateInterest(openingBalance, year));
    }
}
```

```
PS D:\hiya kuliah\P_Daspro\java>  d:; cd '
dt_socket,server=n,suspend=y,address=local
0cfaa701\redhat.java\jdt_ws\java_188f3e68\
Enter the opening balance: 3500000
Enter the duration of saving (years): 12
Amount of money after 12 years: 12244577
```

Questions!

1. Recursive function is a function that calls the function itself

```
static int factorialRecursive(int n)
{
    if (n == 0)
    {
        return (1);
    }
    else
    {
        return (n * factorialRecursive(n - 1));
    }
}
```

2.
3. On the factorialRecursive, the program uses recursive to loop the program, which will call the function again to redo the program, while factioralIterative uses for loop to loop the program, so it will use the basic way by counting the I number until the limit that being given
4. The function is executed by the y times, or based on the main by the exponent times. This happens because the program is being called again by y – 1 times each time the program is being called, and after the y is 0 the program will be terminated
5. Base case

```
if (year == 0)
{
    return (balance);
}
```

Recursion call

```
else
{
    return (1.11 * calculateInterest(balance, year - 1));
}
```

Assignment

1.

```java
import java.util.Scanner;
public class jb11assignment1
{
    static int RecursiveDescendingSeries(int x)
    {
        if (x == 0)
        {
            return (0);
        }
        else
        {
            System.out.print(x);
            return RecursiveDescendingSeries(x - 1);
        }
    }

    static int IterativeDescendingSeries(int x)
    {
        for(int i = x; i > 0; i--)
        {
            System.out.print(x);
            x--;
        }
        return x;
    }

    Run | Debug
    public static void main(String[] args)
    {
        Scanner input = new Scanner (System.in);
        System.out.print(s: "Enter number: ");
        int a = input.nextInt();
        System.out.print(RecursiveDescendingSeries(a));
        System.out.println(x: "");
        System.out.print(IterativeDescendingSeries(a));
    }
}
```

```
Enter number: 5
543210
543210
```

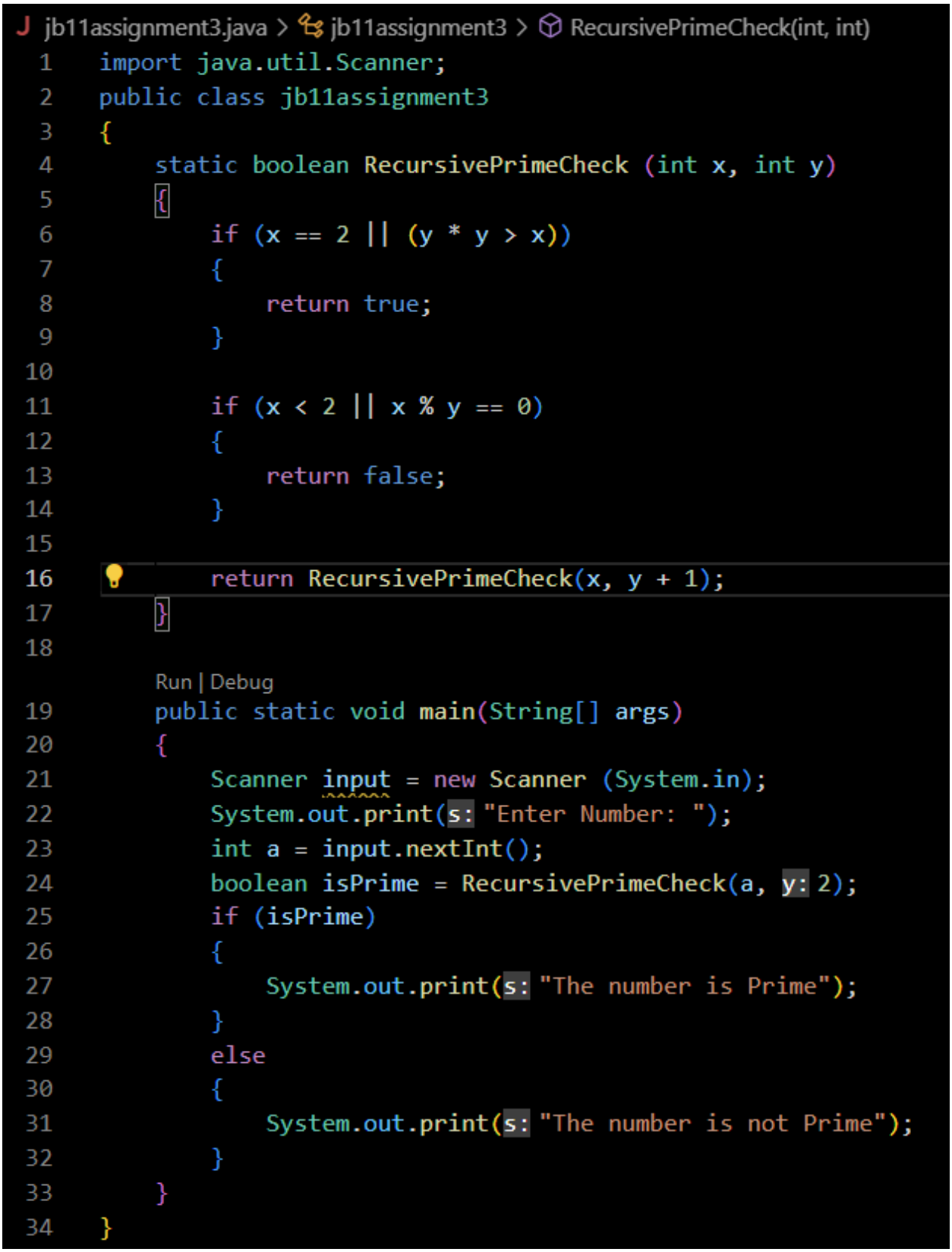
2.

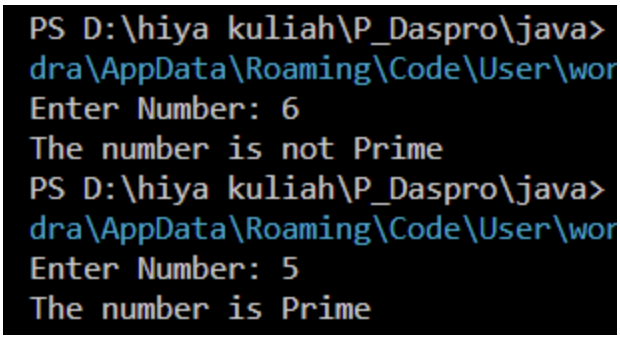
```java
import java.util.Scanner;
public class jb11assignment2
{
    static int RecursiveAddition (int x)
    {
        if (x == 0)
        {
            return (0);
        }
        else
        {
            return (x + RecursiveAddition(x-1));
        }
    }

    Run | Debug
    public static void main (String[] args)
    {
        Scanner input = new Scanner (System.in);
        System.out.print(s: "Enter number: ");
        int a = input.nextInt();
        System.out.print(RecursiveAddition(a));
    }
}
```

```
PS D:\hiya kuliah\P_Daspro\java>
dra\AppData\Roaming\Code\User\wor
Enter number: 5
15
PS D:\hiya kuliah\P_Daspro\java>
dra\AppData\Roaming\Code\User\wor
Enter number: 10
55
```

3.


jb11assignment3.java > jb11assignment3 > RecursivePrimeCheck(int, int)
import java.util.Scanner;
public class jb11assignment3
{
static boolean RecursivePrimeCheck (int x, int y)
{
if (x == 2 || (y * y > x))
{
return true;
}
if (x < 2 || x % y == 0)
{
return false;
}
return RecursivePrimeCheck(x, y + 1);
}
Run | Debug
public static void main(String[] args)
{
Scanner input = new Scanner (System.in);
System.out.print(s: "Enter Number: ");
int a = input.nextInt();
boolean isPrime = RecursivePrimeCheck(a, y: 2);
if (isPrime)
{
System.out.print(s: "The number is Prime");
}
else
{
System.out.print(s: "The number is not Prime");
}
}
}


PS D:\hiya kuliah\P_Daspro\java>
dra\AppData\Roaming\Code\User\wor
Enter Number: 6
The number is not Prime
PS D:\hiya kuliah\P_Daspro\java>
dra\AppData\Roaming\Code\User\wor
Enter Number: 5
The number is Prime