JPL :: Writing Efficient Programs

TalentSprint

Licensed To Skill

Version 1.0.4

Learning Objectives

The content in this presentation is aimed at teaching learners to:

- Provide alternative solutions to the same problem
- Optimize solutions to problems
- Write elegant and structured code for problems
- Write programs to problems by decomposing functionality into methods and using the methods

Writing Efficient Programs

Let’s re-look at the solution to find Prime Numbers

Solution

```
Read the number into n.
for i from 2 to n−1,
    if n % i = 0, then print ("n not prime").
        print("n prime");
```

Alternatively, we can also Count the number of divisors of the given number. If it is more than 2, it is not a prime number. Else, it is a prime number.

Writing Efficient Programs

Here is an alternative solution

```
Read the number into n.
for i from 1 to n,
    if n % i = 0, then increment count
if count is 2, Print ("n prime");
else print ("n not prime")
```

Now, let us write Java code for the same.

Writing Efficient Programs

```
public class PrimeNumber2 {
    public static void main(String[] args) {
        int i;
        int n = Integer.parseInt(args[0]);
        int factorCount = 2;
        for (i = 2; i <= n - 1; i++) {
            if (n % i == 0) {
                factorCount++;
            }
        }
        if (factorCount == 2) {
            System.out.println(n + " is prime");
        } else {
            System.out.println(n + " is not prime.");
        }
    }
}
```

Writing Efficient Programs

Code

```
for (i = 2; i <= n - 1; i++)
    if (n % i == 0)
        factorCount++;
if (factorCount == 2)
    System.out.println(n + " is prime");
else
    System.out.println(n + " is not prime.");
```

$\downarrow$

Instead of ‘n-1’ why not ‘n/2’ !

Writing Efficient Programs

Solution

```
public class PrimeNumber3 {
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        int factorCount = 2;
        for (int i  = 2; i  <= n / 2; i++) {
            if  (n % i == 0) {
                factorCount++;
            }
        }
        if  (factorCount == 2) {
            System.out.println(n + " is prime");
        }  else
            System.out.println(n + " is not prime.");
    }
}
```

Writing Efficient Programs

Do we really need to loop thru 'n/2' ?

Can we do better?

How about sqrt(n)? Is it sufficient? If yes, why?

Writing Efficient Programs

Solution

```
public class PrimeNumber3 {
    public static void main(String[] args) {
        int n = Integer.parseInt(args[0]);
        int factorCount = 2;
        for (int i  = 2;  i  <= Math.sqrt(n); i++) {
            if  (n % i == 0) {
                factorCount++;
            }
        }
        if  (factorCount == 2) {
            System.out.println(n + " is prime");
        }  else
            System.out.println(n + " is not prime.");
    }
}
```

Writing Efficient Programs

Our earlier solution for finding Perfect Square

```
public class PerfSquare {
    public static void main(String[] args) {
        int i = 1;
        int givenNumber = Integer.parseInt(args[0]);
        while (i < givenNumber) {
            if (i * i == givenNumber) {
                System.out.println(givenNumber + " is perfect
  square.");
                return;
            }
            i++;
        }
        System.out.println(givenNumber + " is not perfect
  square.");
  }
```

Writing Efficient Programs

A Better Solution for Finding Perfect Square:

```
public class PerfSquare {
    public static void main(String[] args) {
        int i = 1;
        int n = Integer.parseInt(args[0]);
        while (i * i < n) {
            i++;
        }
        if (i * i == n)
            System.out.println(n + " is perfect square.");
        else
            System.out.println(n + " is not perfect square.");
    }
}
```

Writing Efficient Programs

1. Write a program for perfect square using `sqrt()` function.
2. Print all perfect squares between 1 and a given number ‘n’.

Writing Efficient Programs

Solution for Finding Perfect Square upto ‘n’:

```
public class PerfSquareRange {
    public static void main(String[] args) {
        int i , j;
        int n = Integer.parseInt(args[0]);
        for (i = 1; i <= n; i++) {
            j = 1;
            while (j * j < i) j++;
            if (j * j == i)
                System.out.println(i + " is perfect square.");
        }
    }
}
```

Writing Efficient Programs