|  |  |
| --- | --- |
| Image result for Worst-case asymptotic analysis  Worst-case asymptotic analysis | Abstract  In computer science, the worst-case complexity (usually denoted in asymptotic notation) measures the resources an algorithm requires in the worst -case.  Roxanne Meeks, Bernard Smith, Rasheed Walcott |

Contents

[Worst Case Asymptotic Analysis 2](#_Toc25531414)

[Admin.java 2](#_Toc25531415)

[Driver.java 5](#_Toc25531416)

[OpeningHours.java 7](#_Toc25531417)

[Places.java 9](#_Toc25531418)

[PlacesNode.java 15](#_Toc25531419)

[PlacesTree.java 16](#_Toc25531420)

[RequestNode.java 20](#_Toc25531421)

[RequestQueue.java 21](#_Toc25531422)

[RequestStack.java 23](#_Toc25531423)

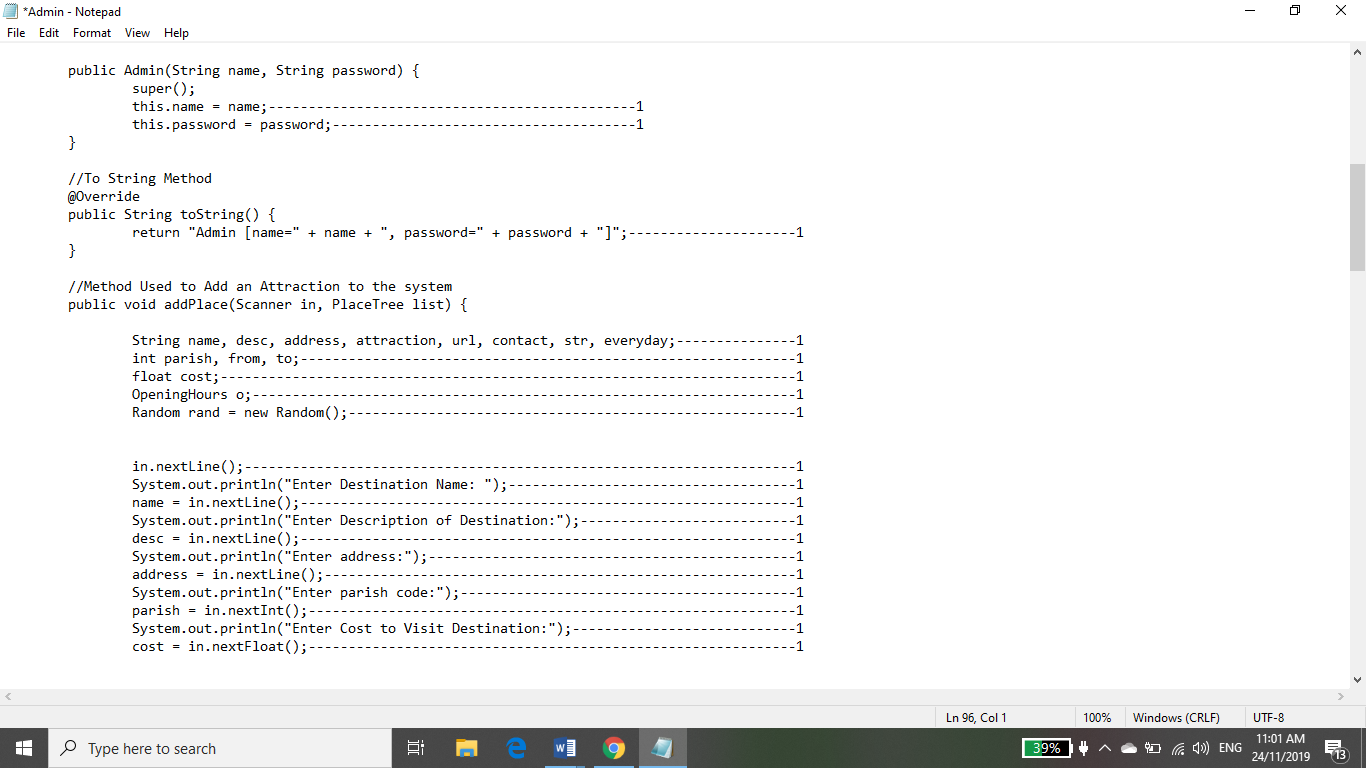
[TreeNode.java 24](#_Toc25531424)

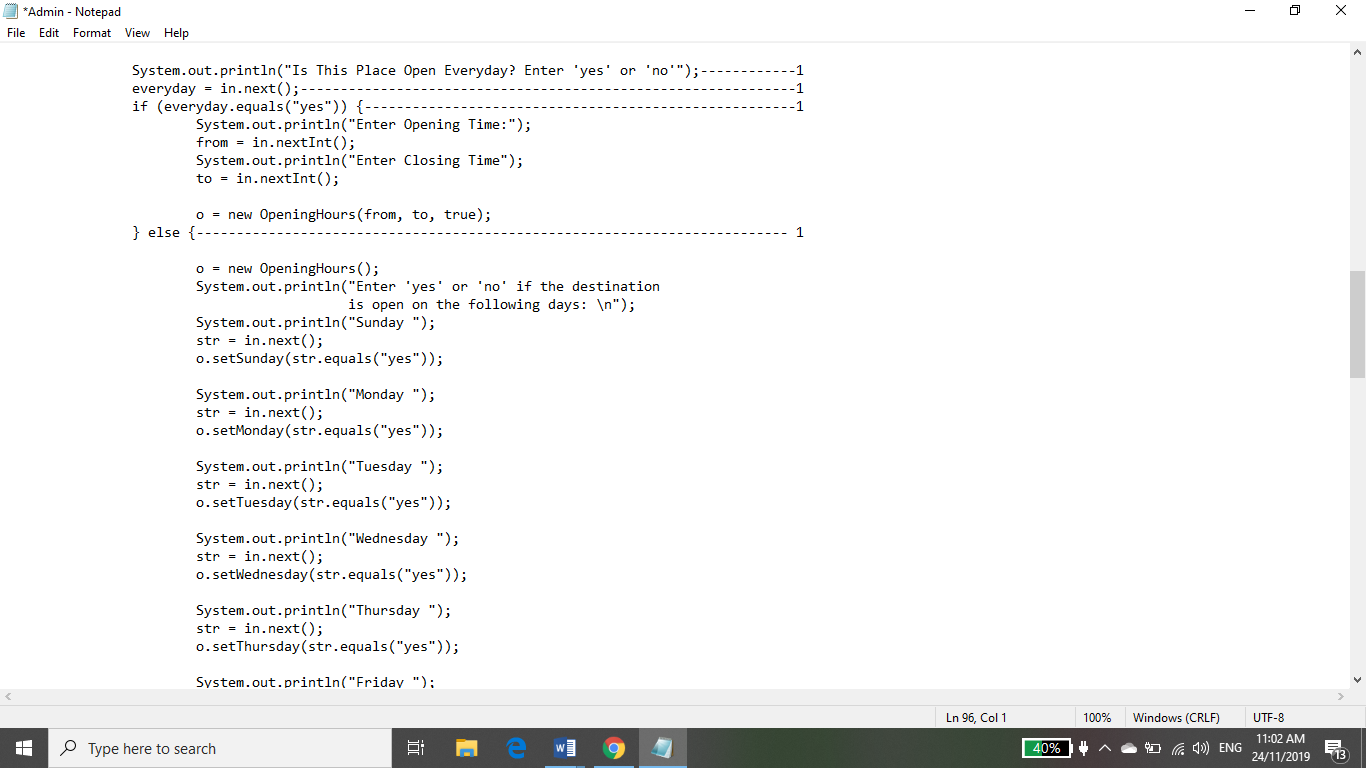
[Visitor.java 25](#_Toc25531425)

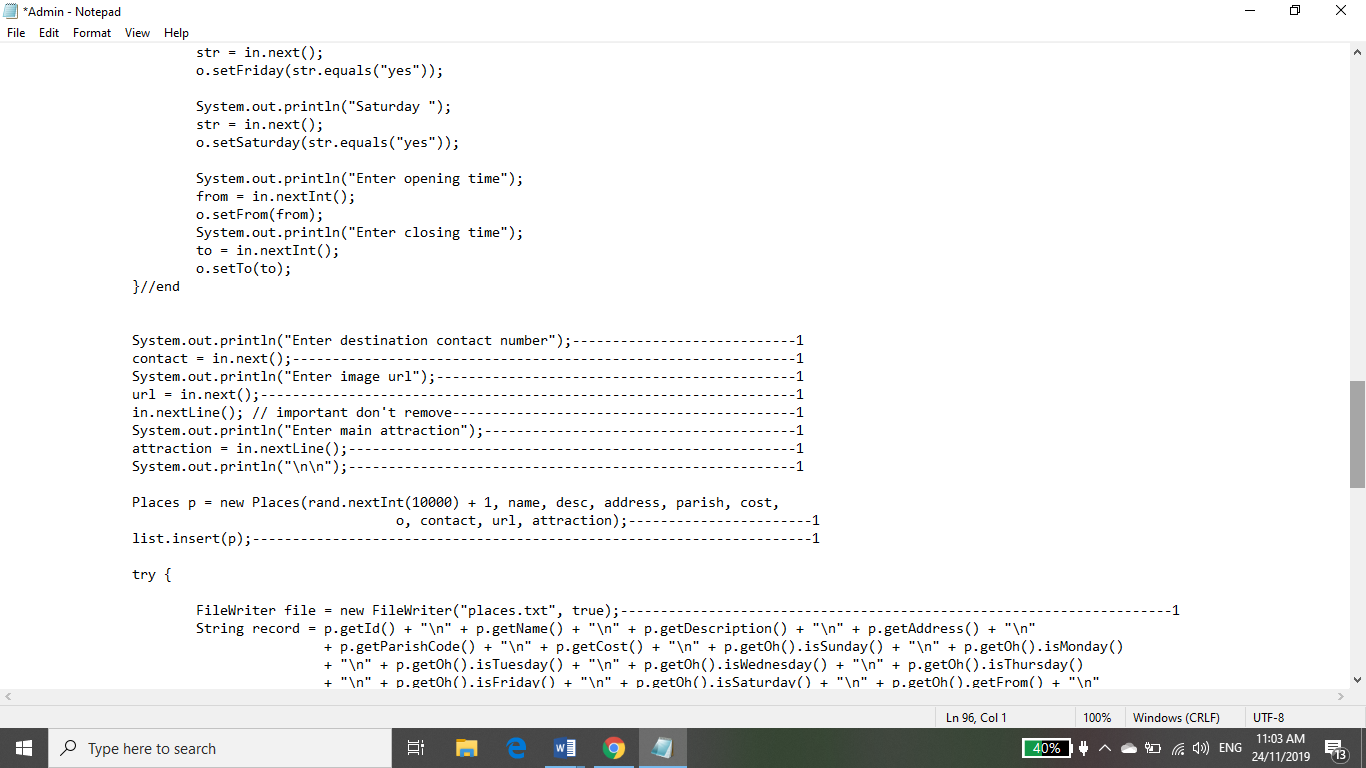
# Worst Case Asymptotic Analysis

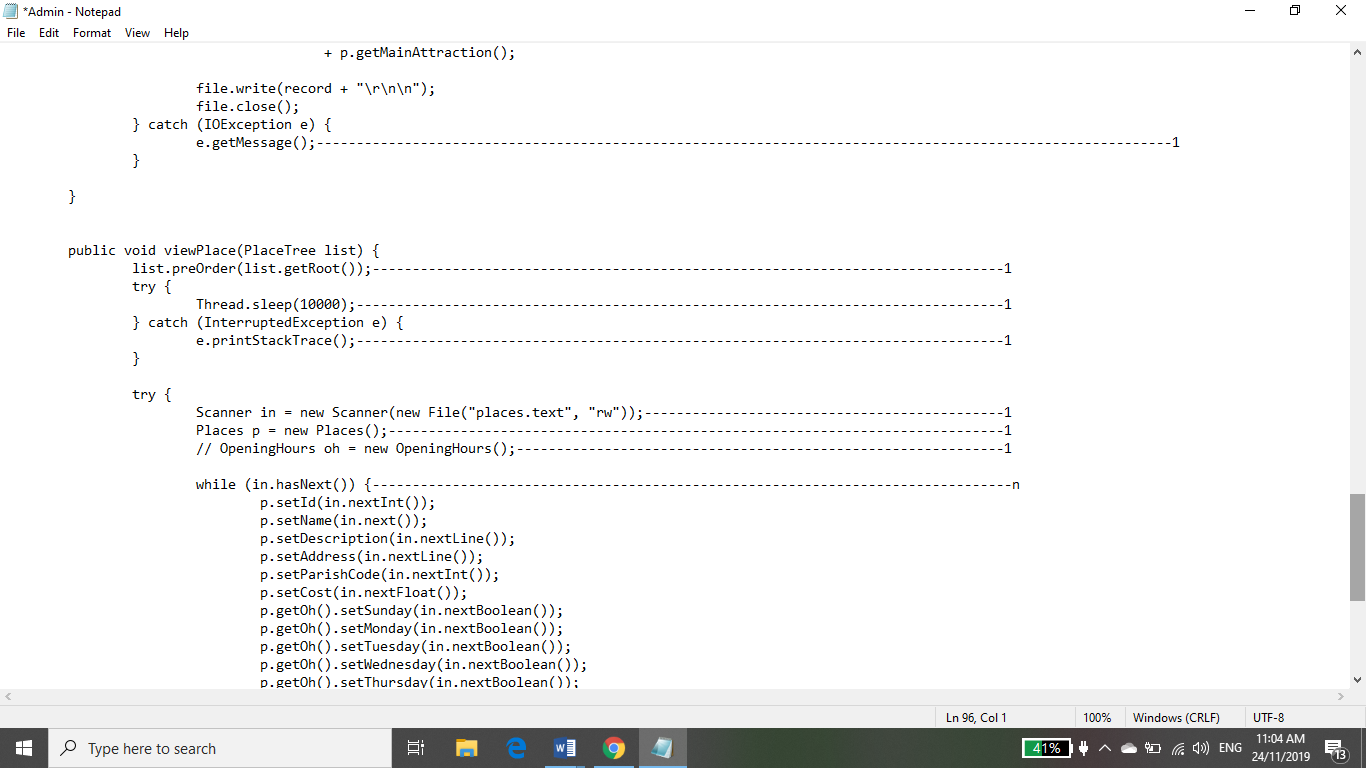
## Admin.java

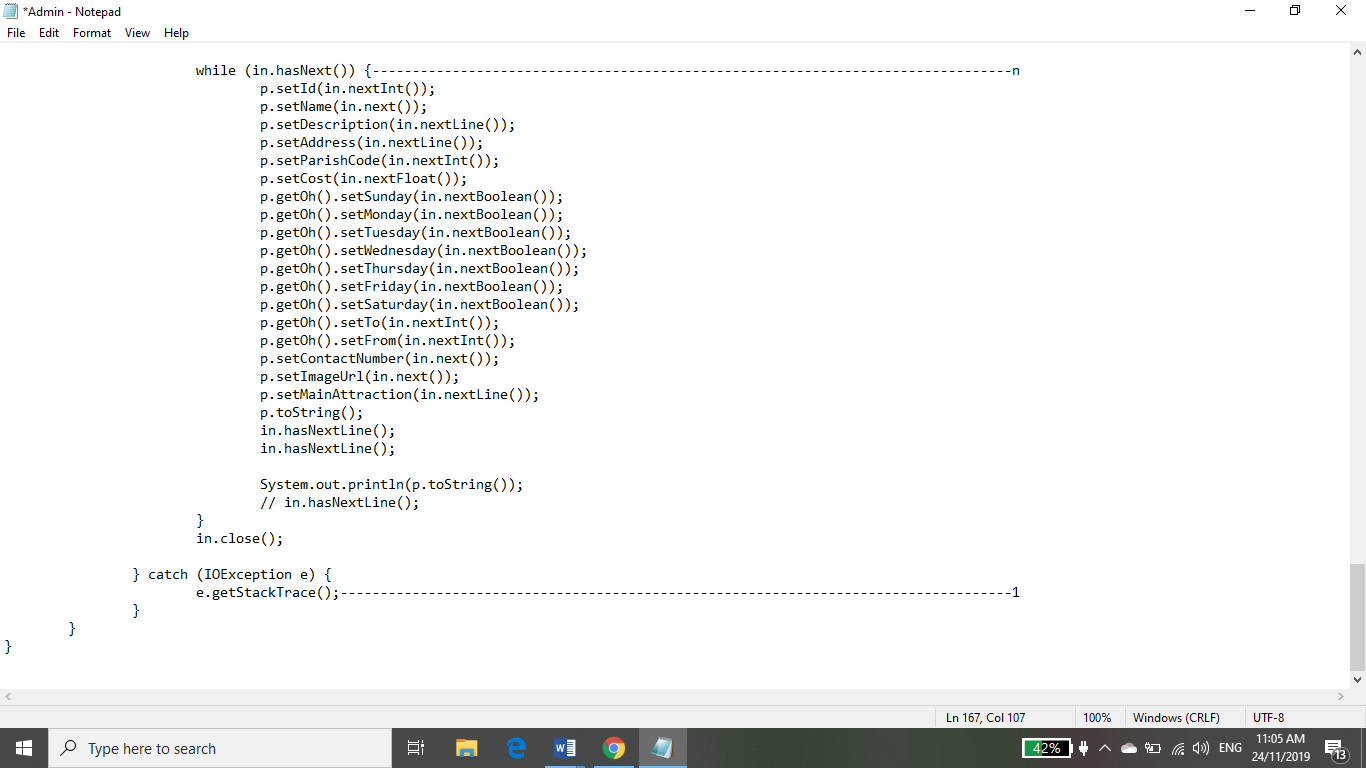












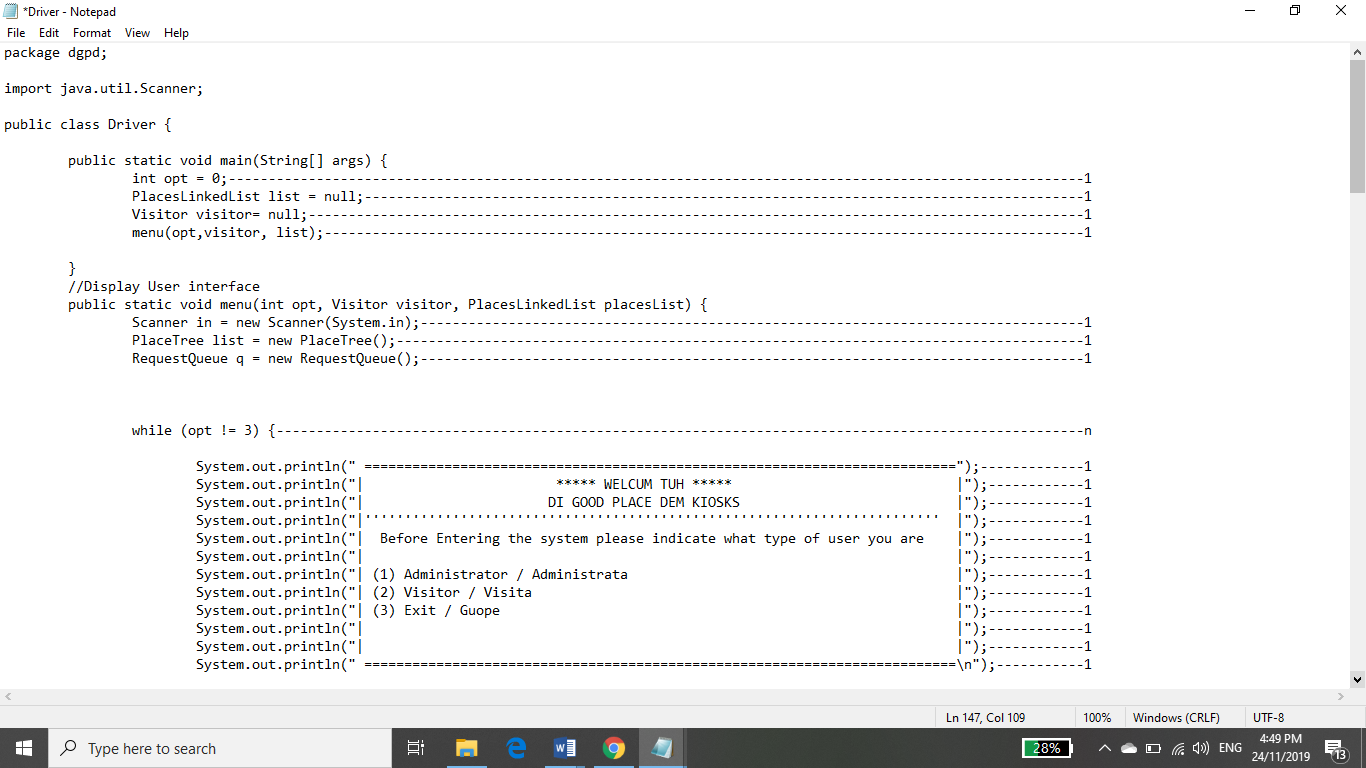
Analysis of Admin.java

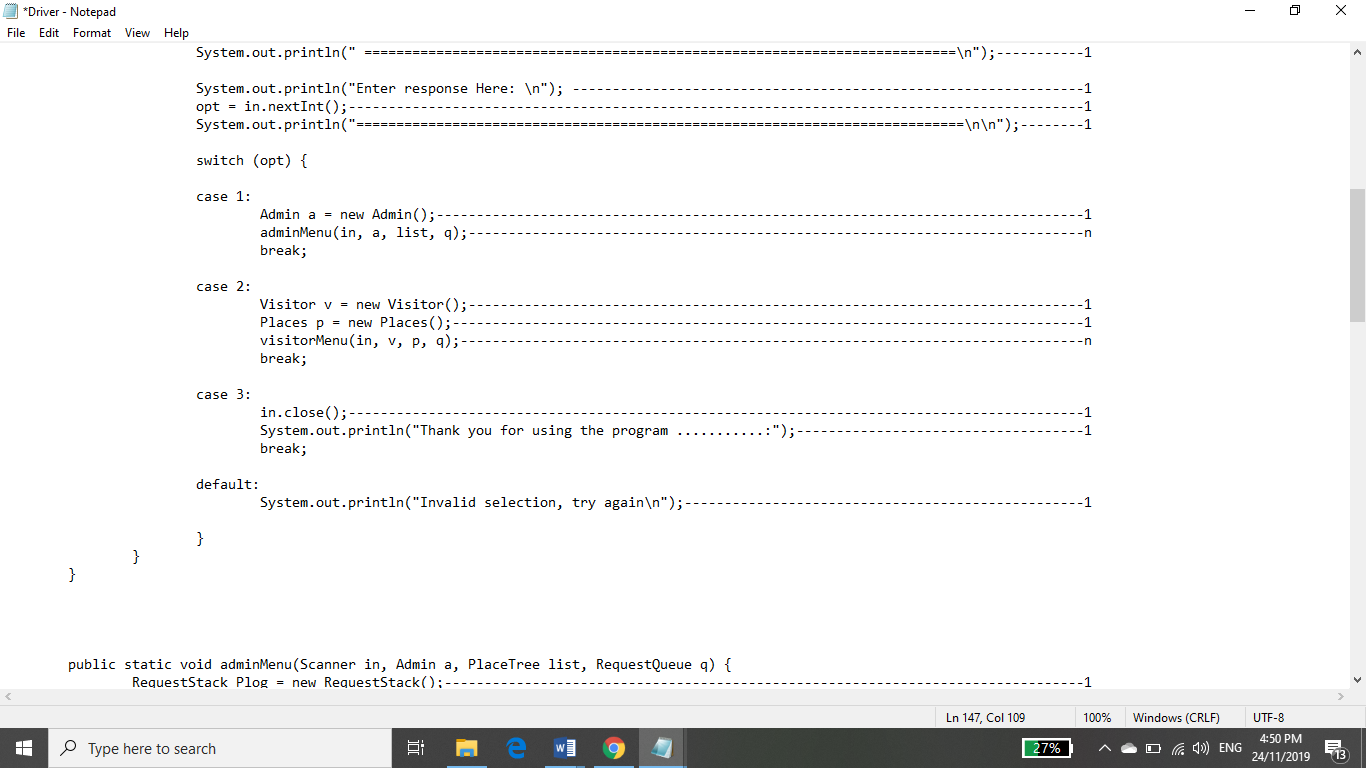
f(n) = 49+n(1)

f(n) = n

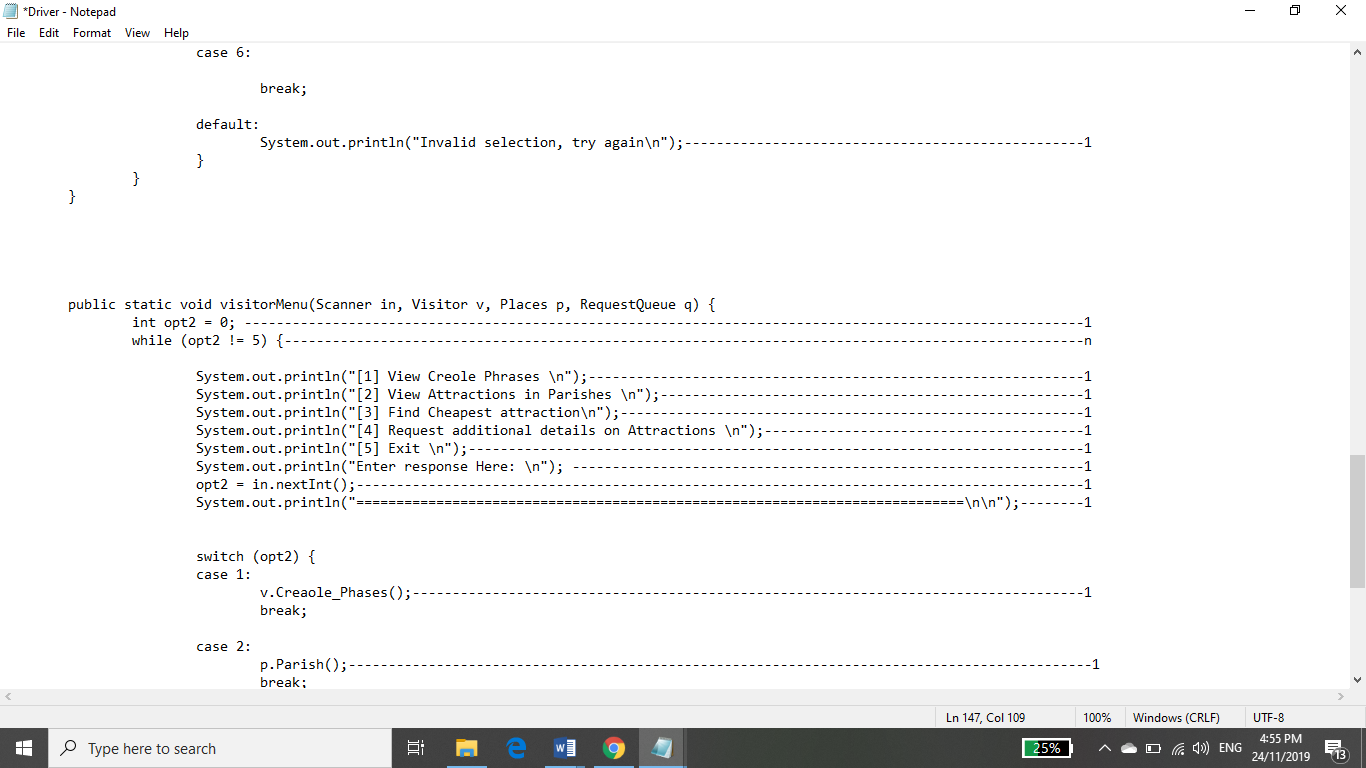
O(f(n)) = n, linear

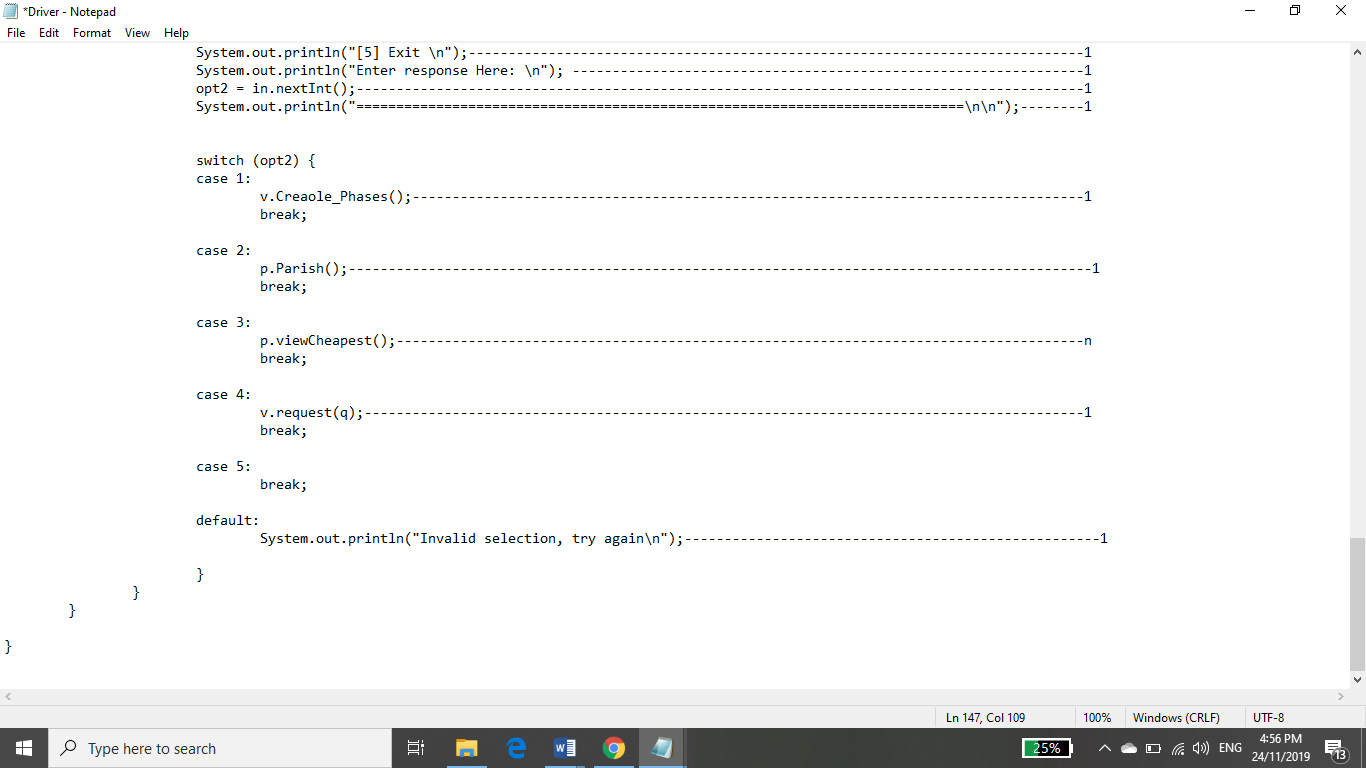
## Driver.java











Analysis of Driver.java

f(n)= 4+(3(n+16(1+n+n+1+n+1)))+ 1(n+8(1+1+n+1+1))))

f(n)= 4+(3(n+16(3n +3)))+ 1(n+4))))

f(n)=4+(3n+48(3n+3))+ n+ 4

f(n)= 8+(3n+144n+ 144)+n

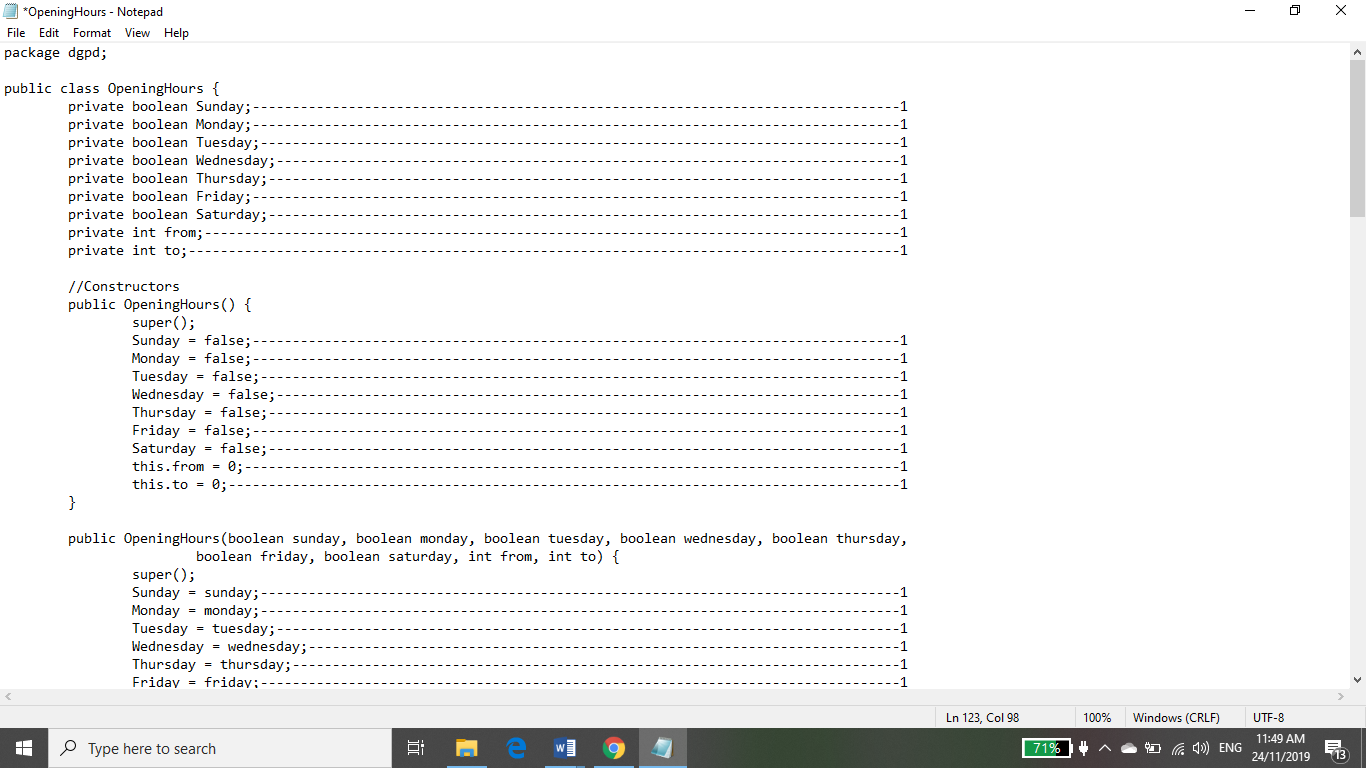
f(n)= 8+144+ 3n+144n+n

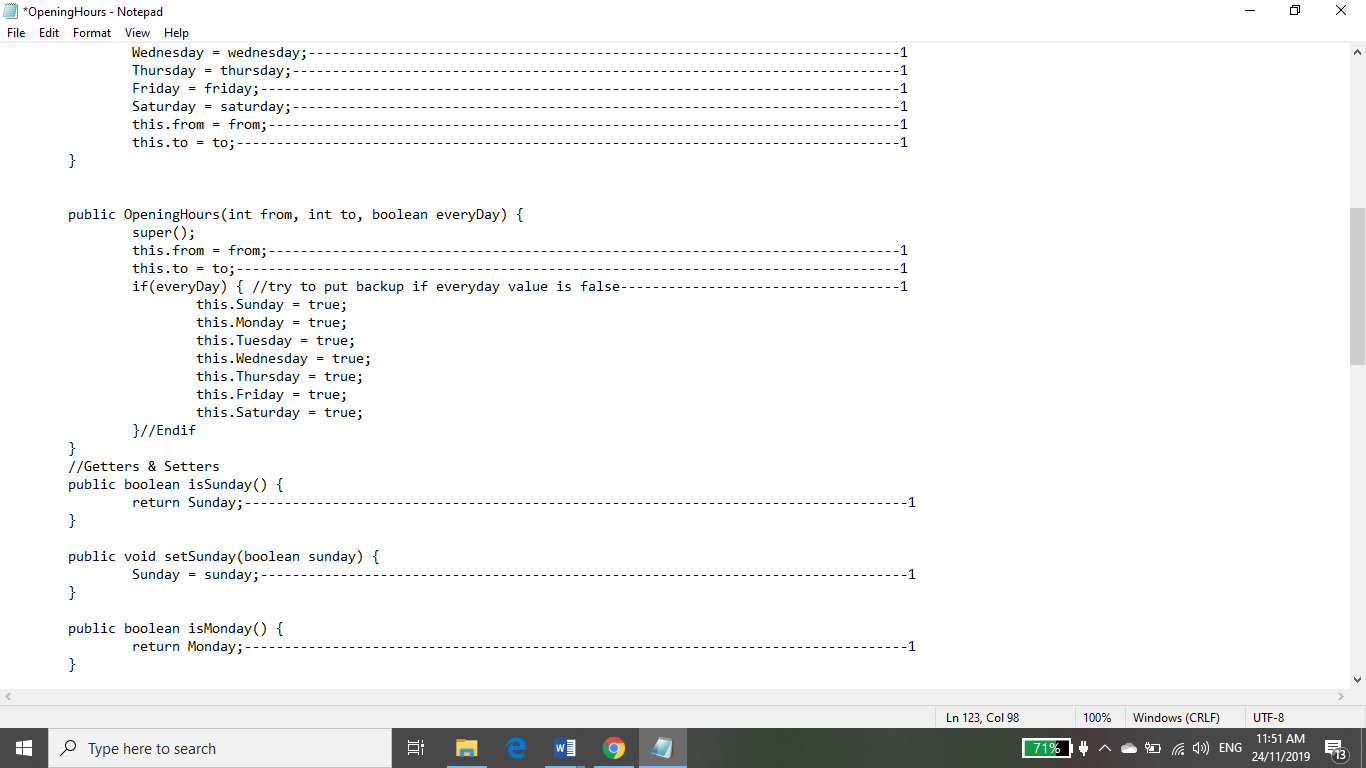
f(n)= 152+ 148n

f(n)= n

O(f(n)) = n, linear

## OpeningHours.java









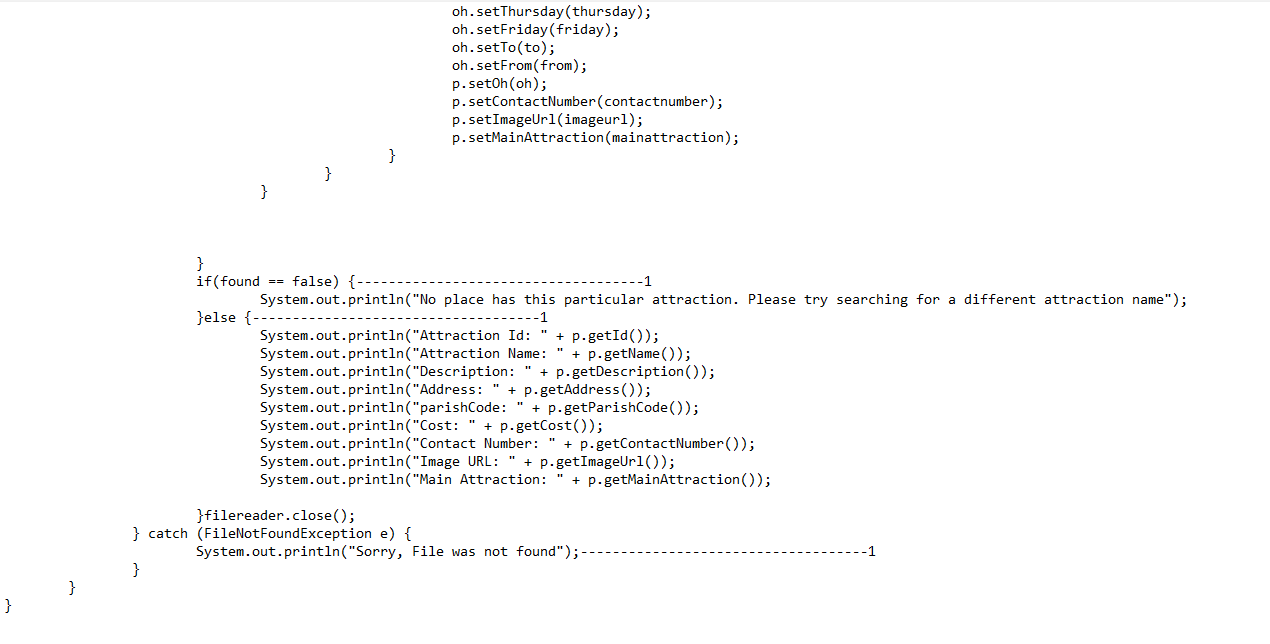
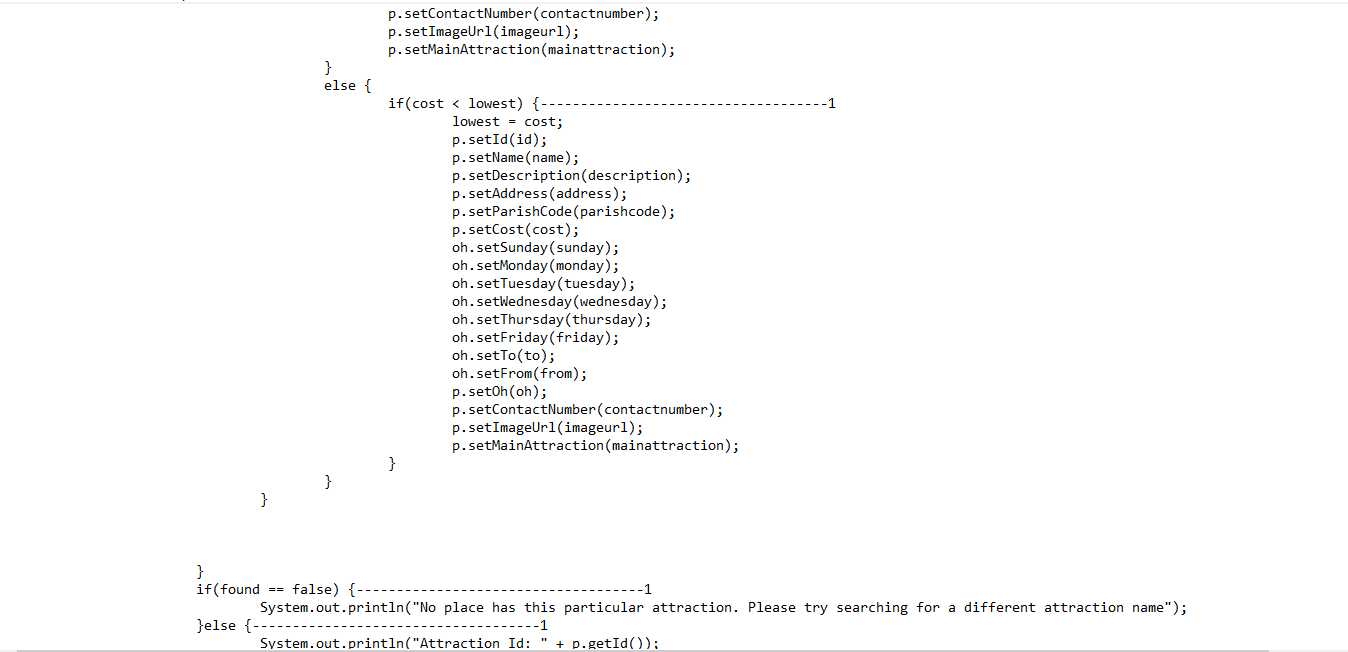
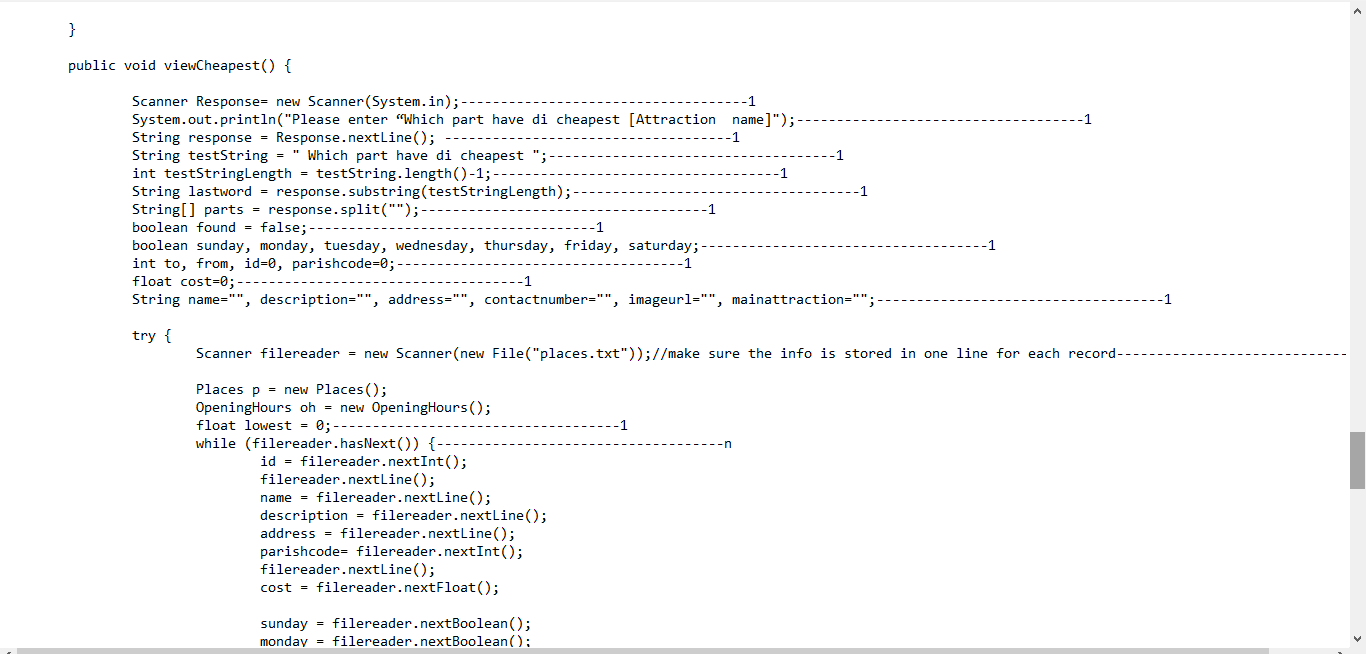
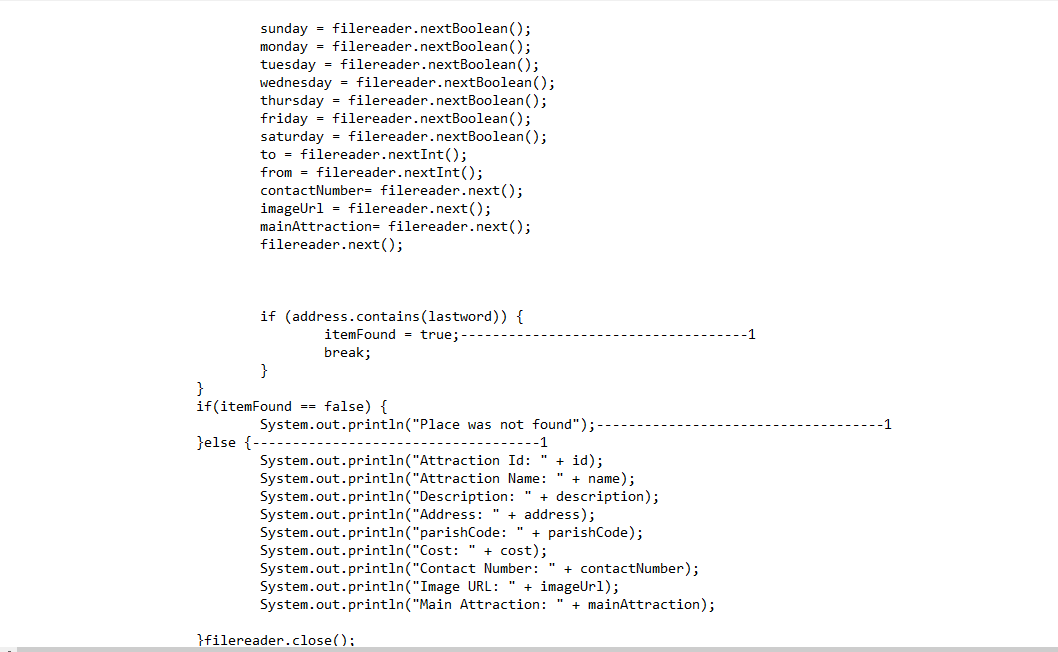
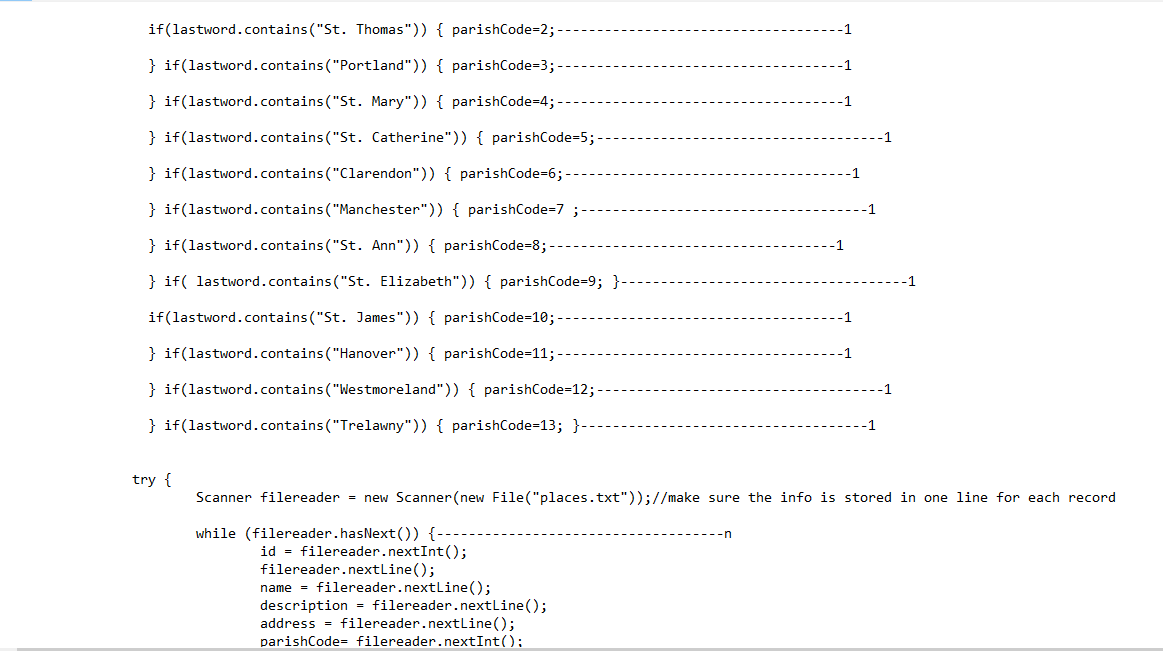
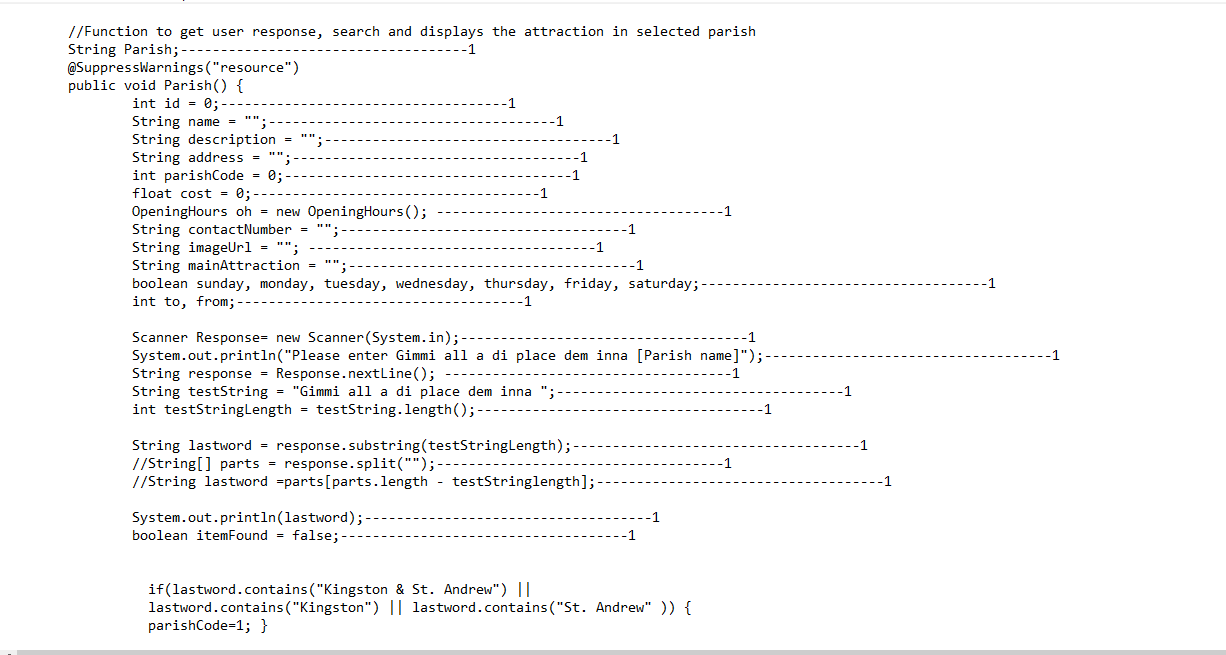
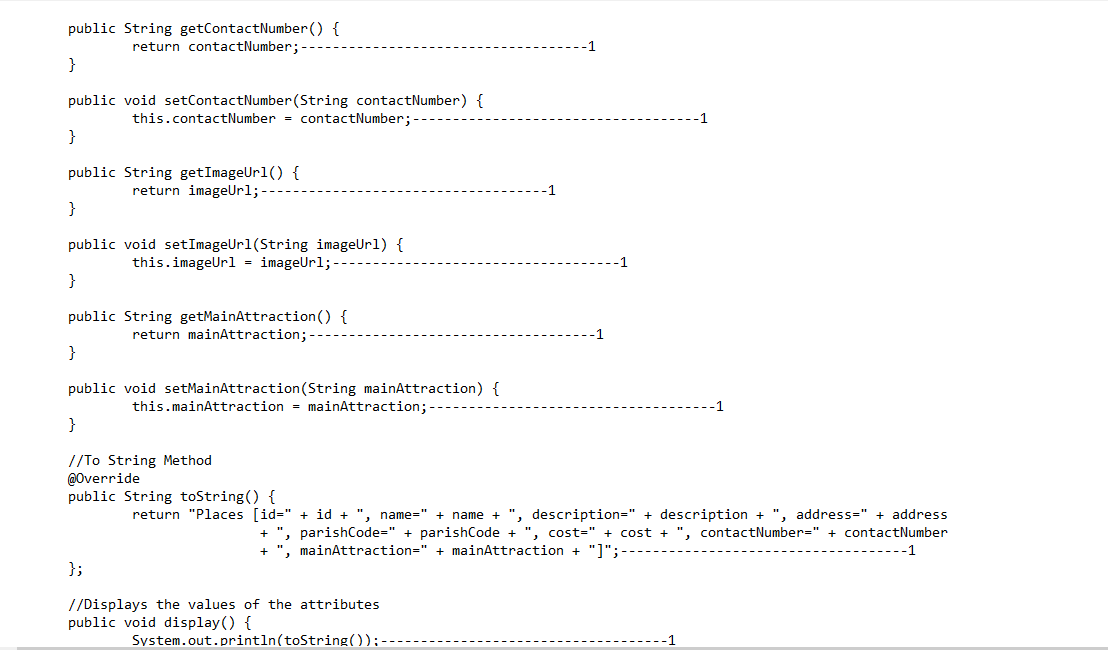
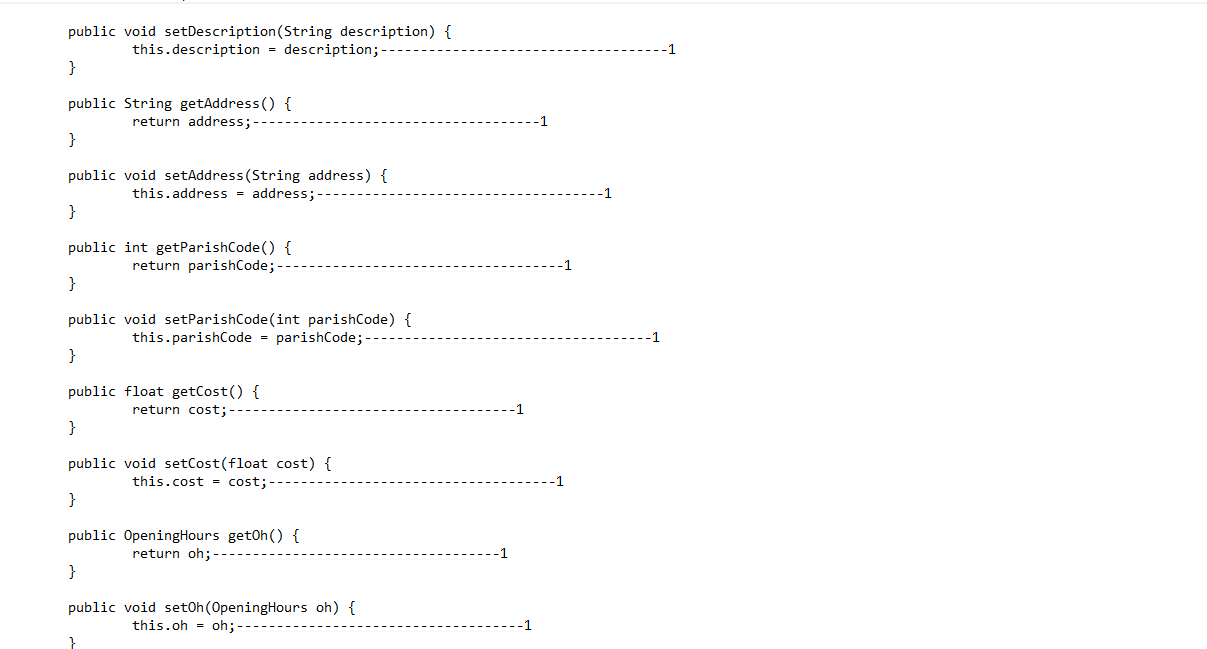
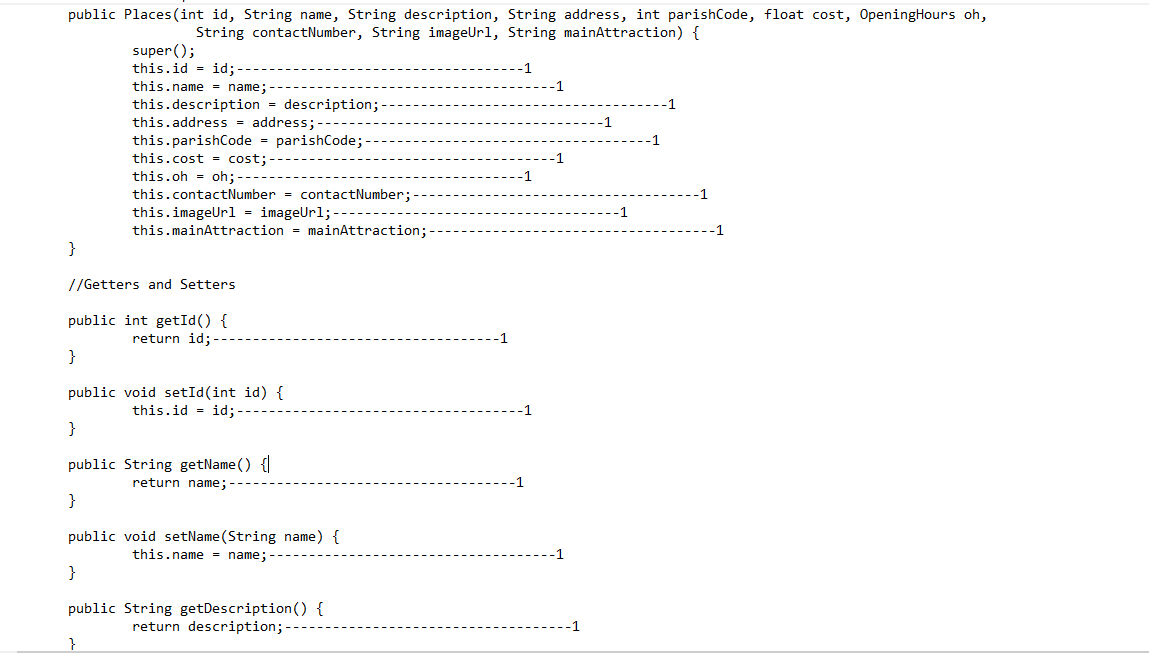
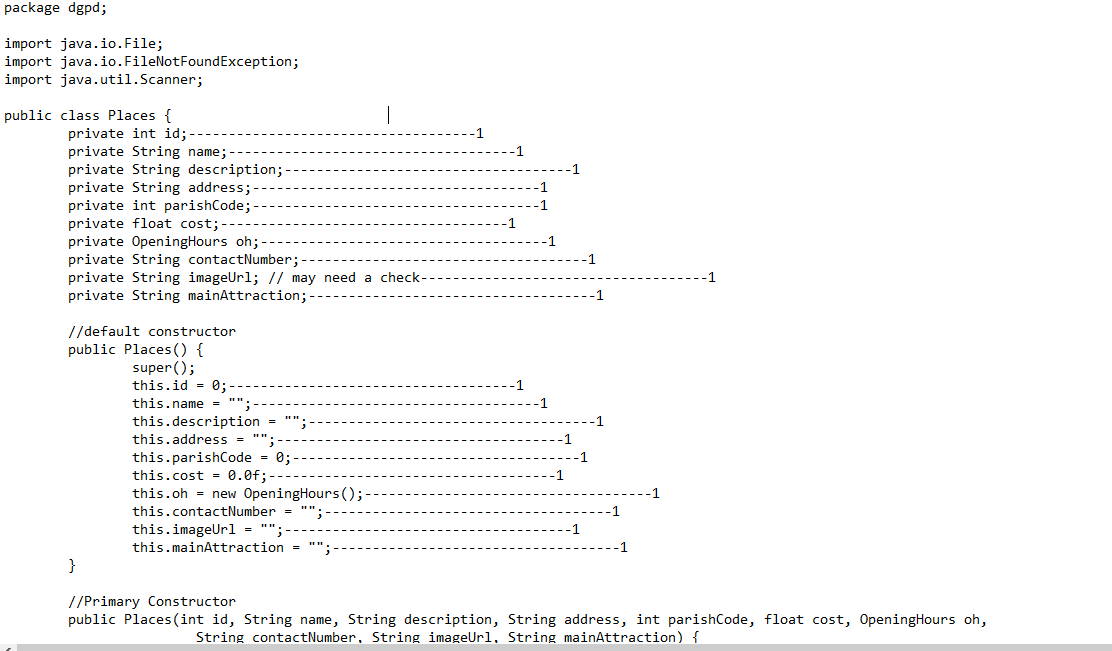
Analysis of OpeningHours.java

f(n) =9+(9)+(9)+(3)+ 18

f(n)= 48

O(f(n)) = 1, Constant

## Places.java



Analysis of Places.java

f(n)=(10)+(10)+(10)+20+(1)+(1)+1+(20)+(1)+12+(n)+(1)+(1)+(1)+(1)+(12)+(2)+(n)+(1)+(1)+(1)+(1)

f(n)=86+n+18+n+4

f(n)=86+18+4+n+n

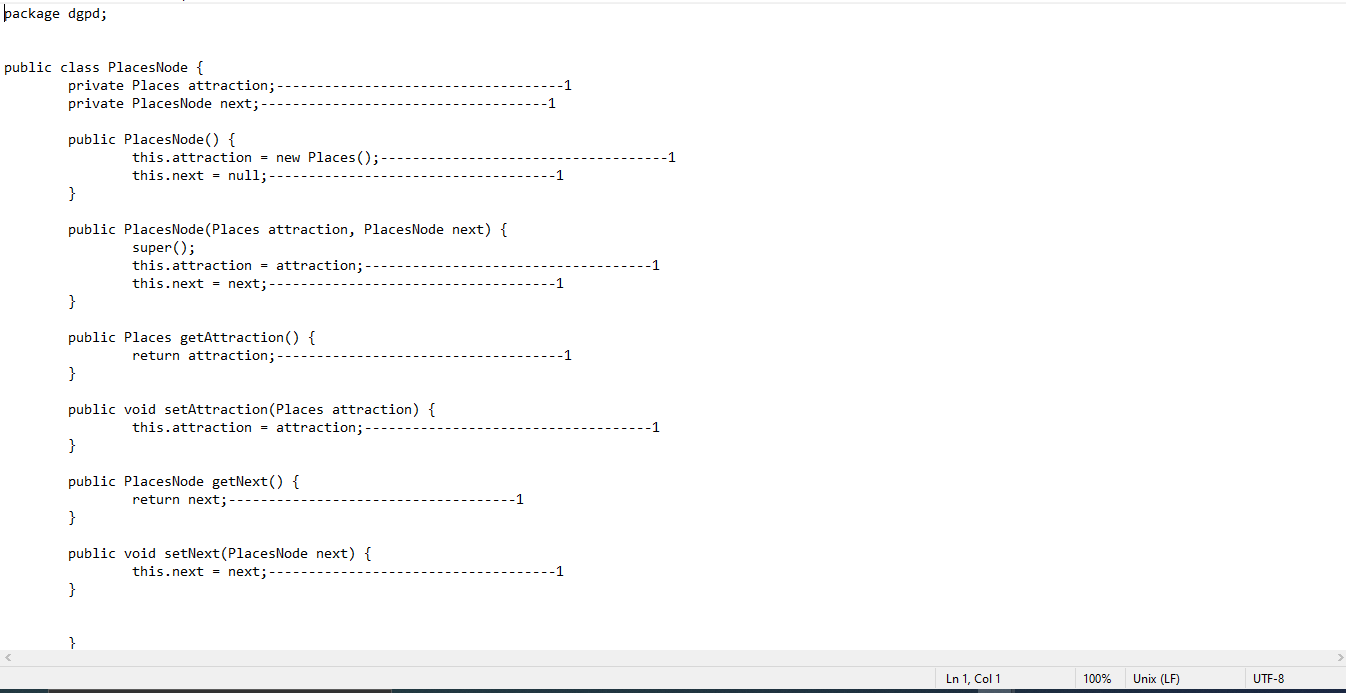
f(n)=108+(2n)

f(n)=2n

f(n)=n

O(f(n))= n, linear

## PlacesNode.java



Analysis of PlacesNode.java

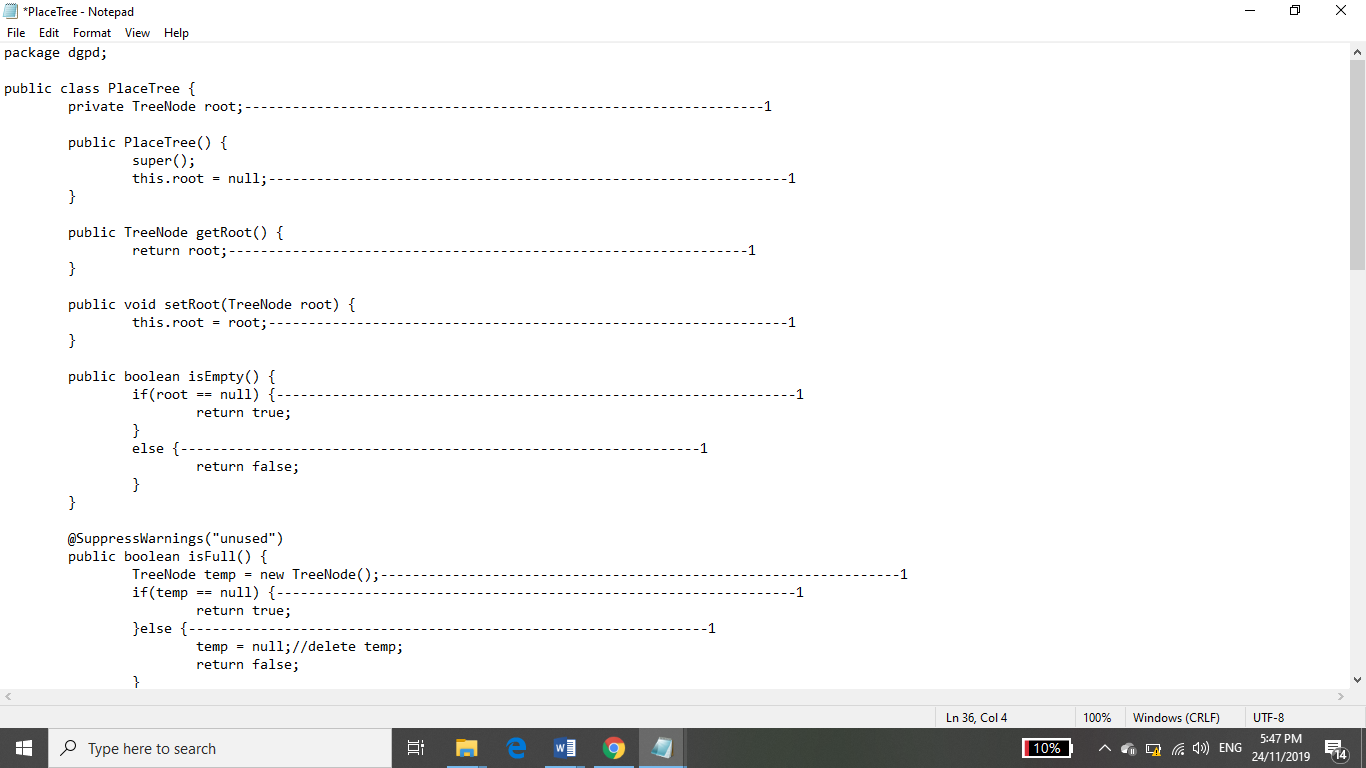
f(n)=1+1+1+1+1+1+1+1+1+1

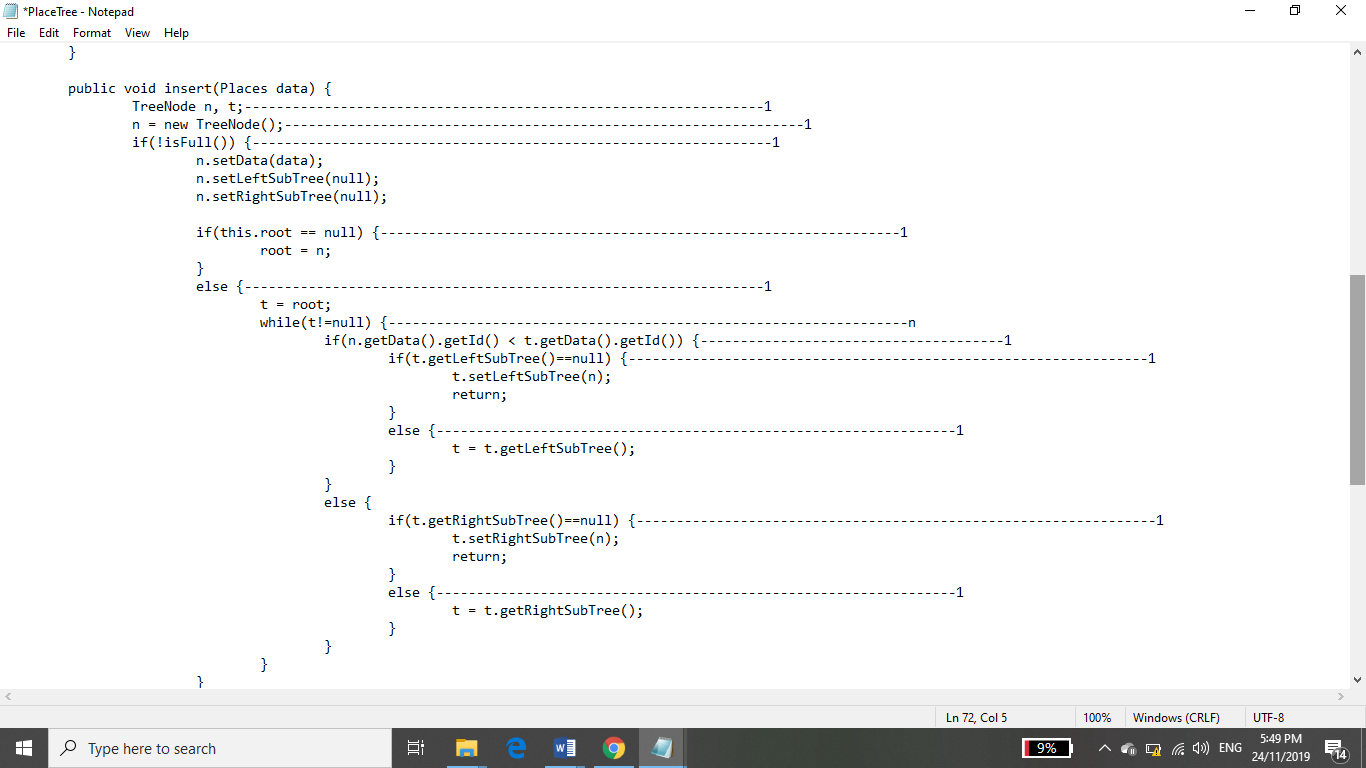
f(n)= 10

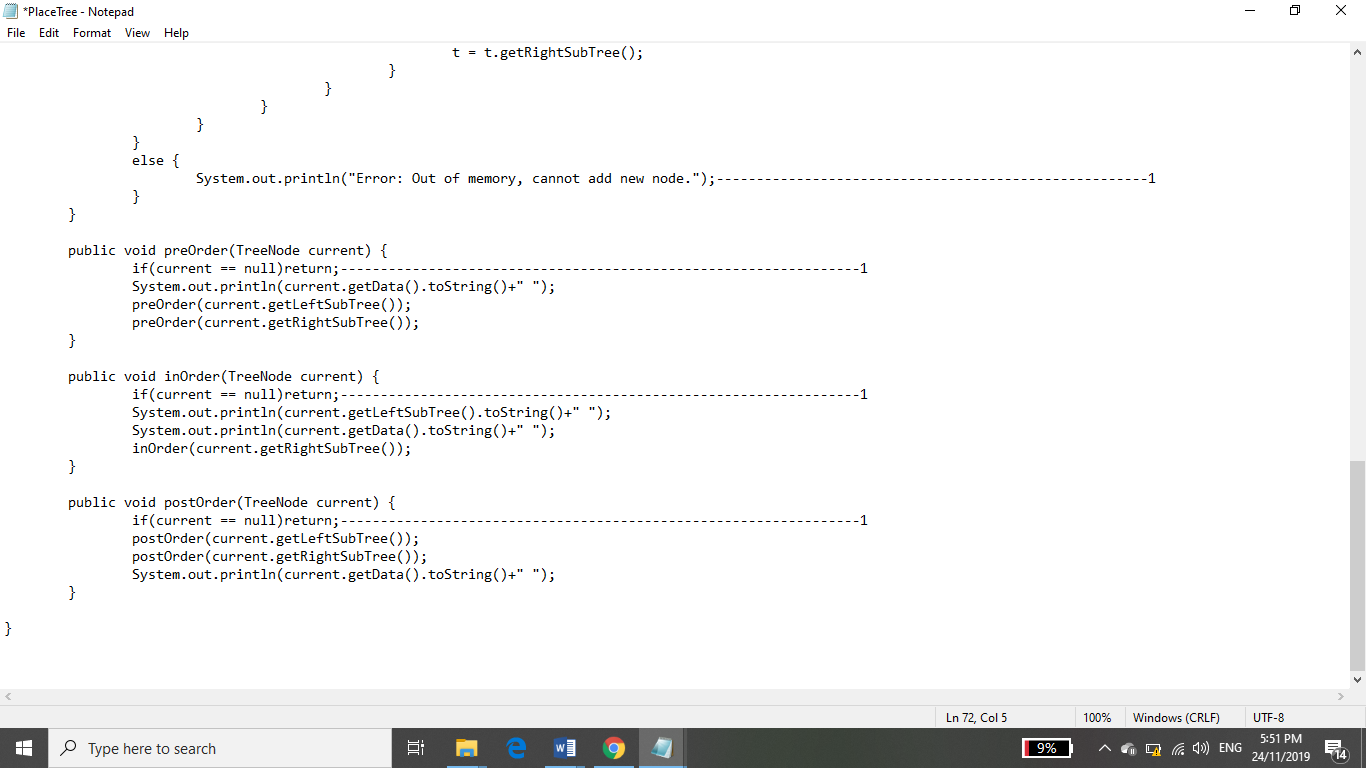
f(n)= 1

O(f(n))= 1, Constant

## PlacesTree.java







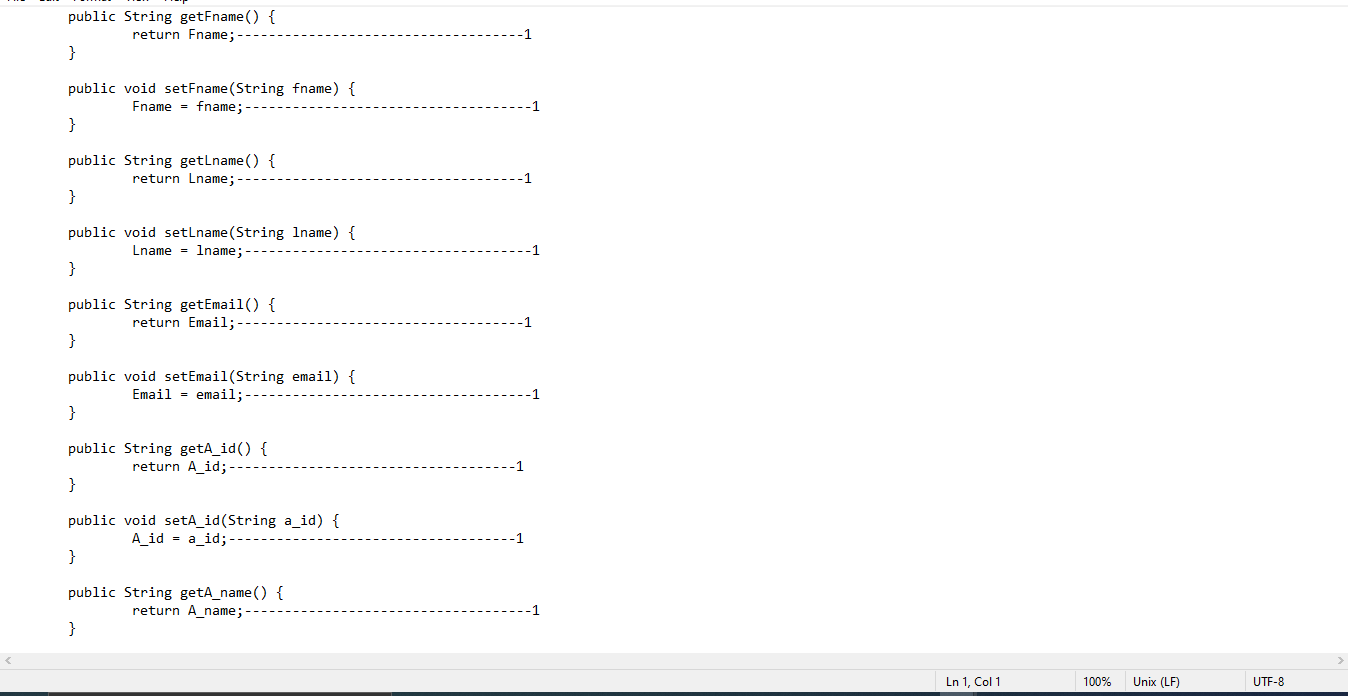
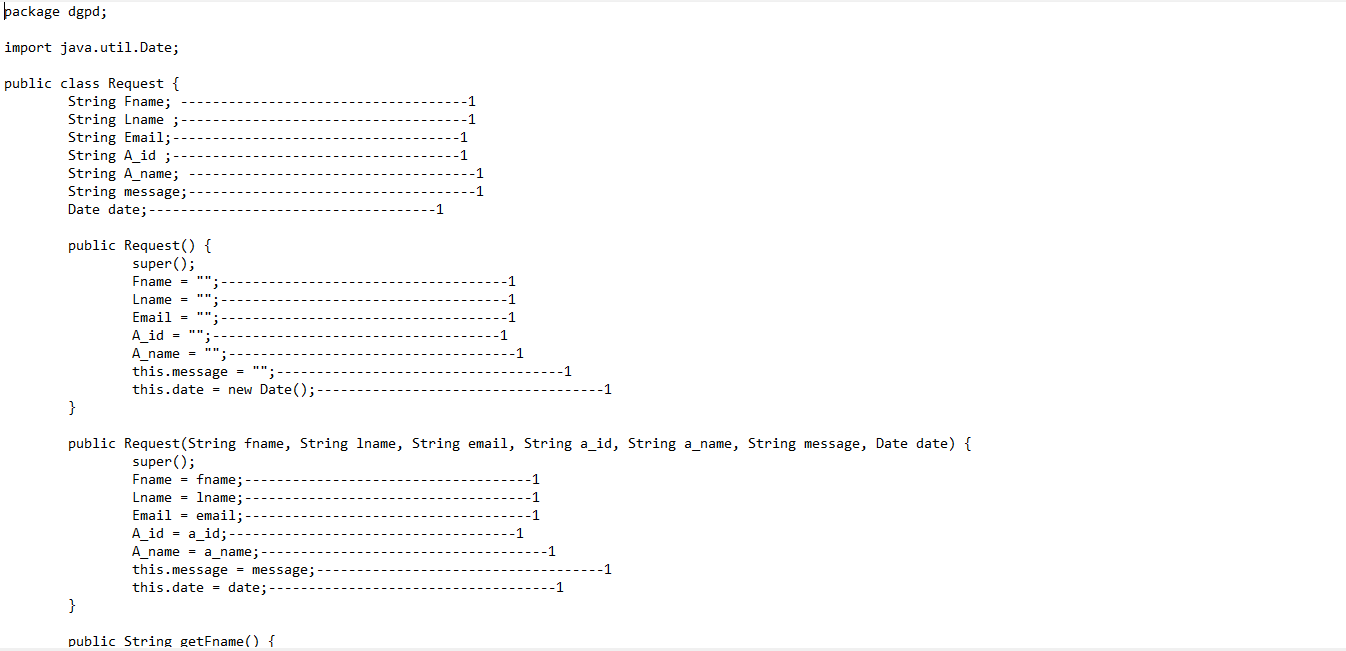
Analysis of PlacesTree.java

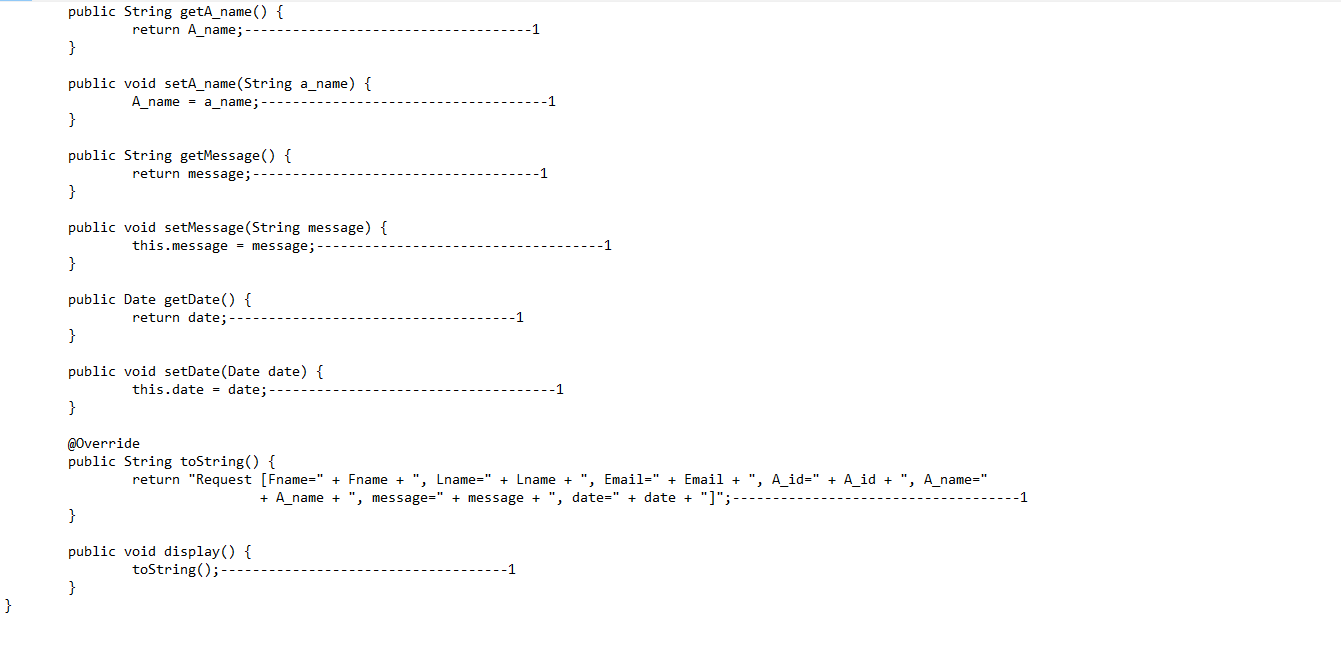
f(n) = 14(n(1)) +7

f(n)= 14n +7

f(n)= n

O(f(n)) = n, linear

Request.java



Analysis of PlacesTree.java

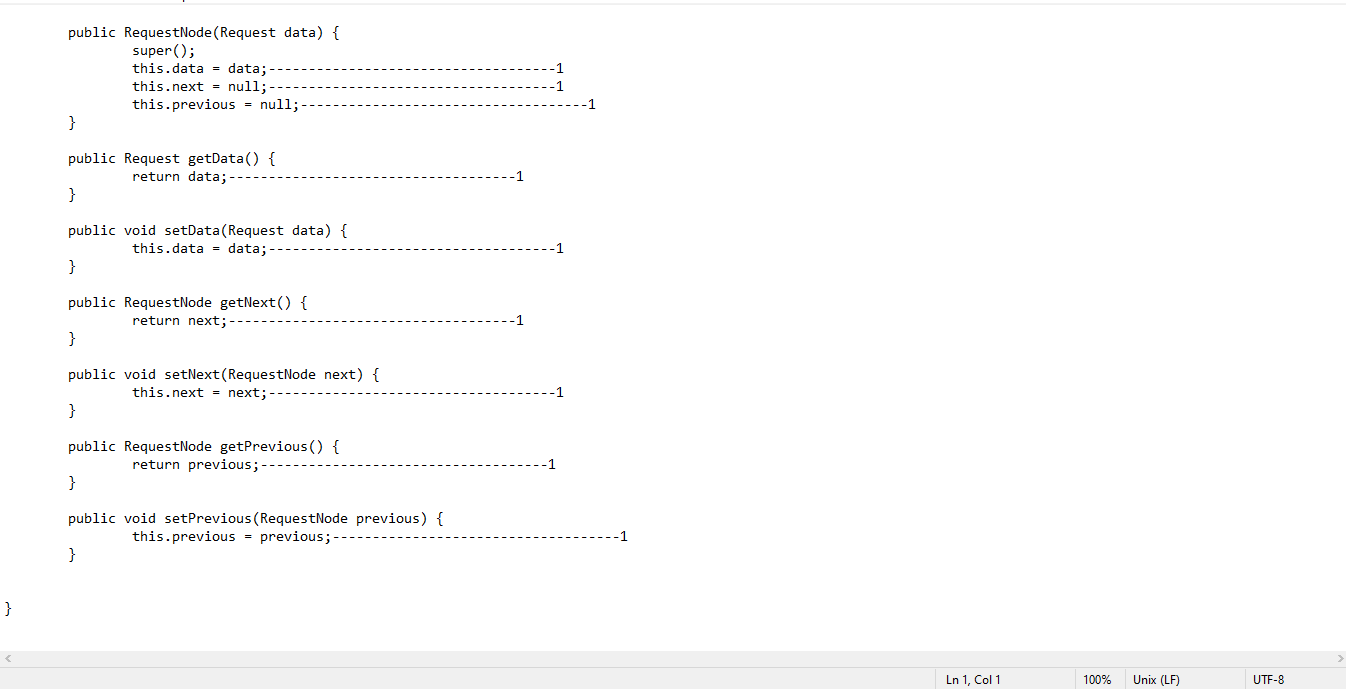
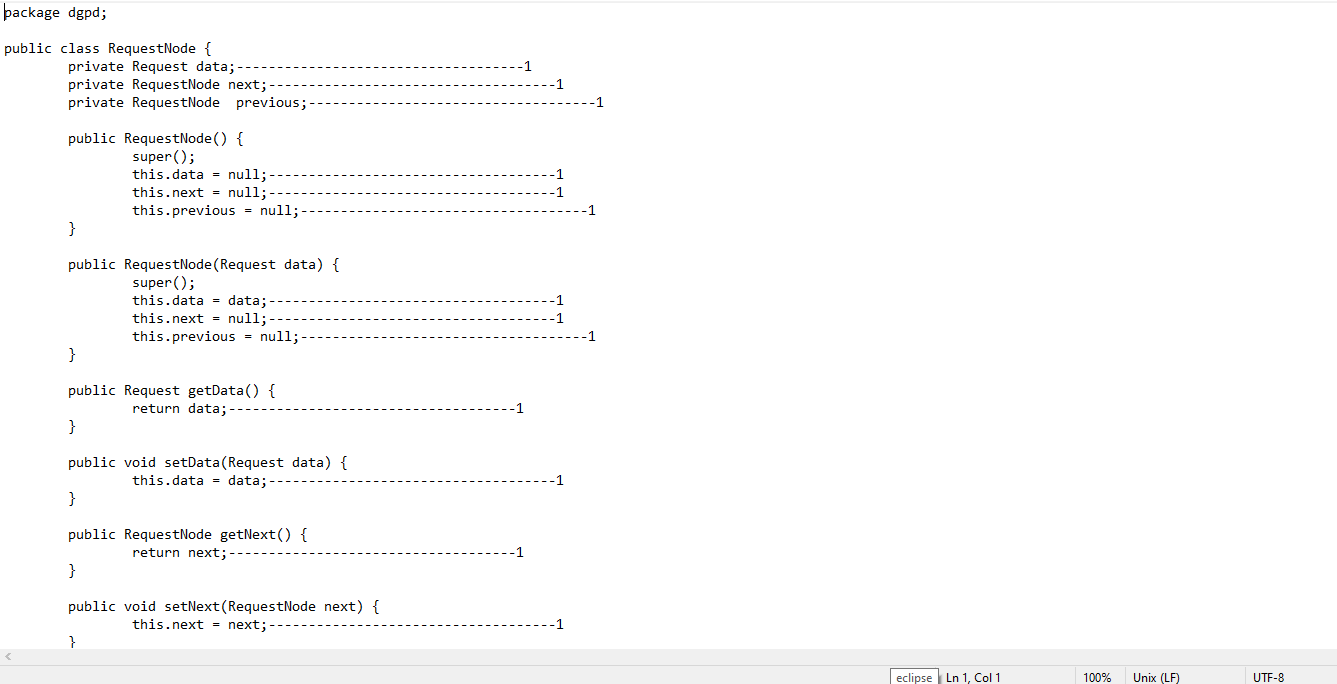
f(n)=7+7+7+ 16

f(n)= 37

f(n)= 1

O(f(n))= 1, Constant

## RequestNode.java



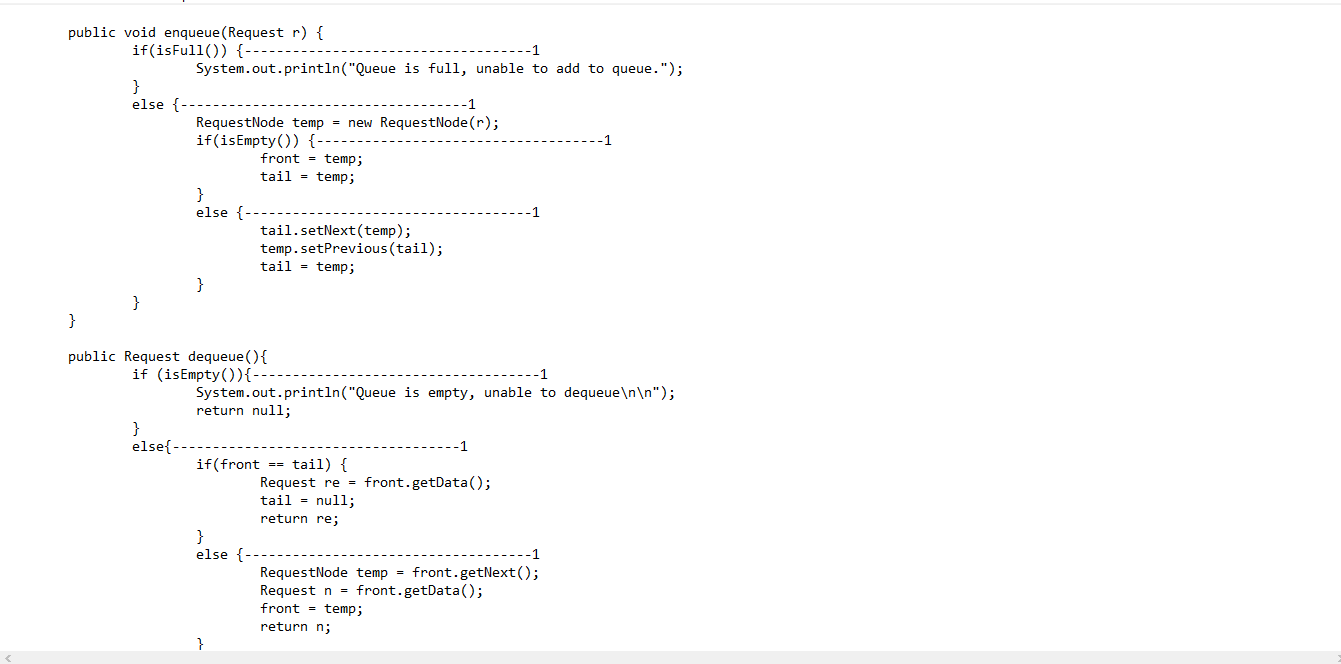
Analysis of RequestNode.java

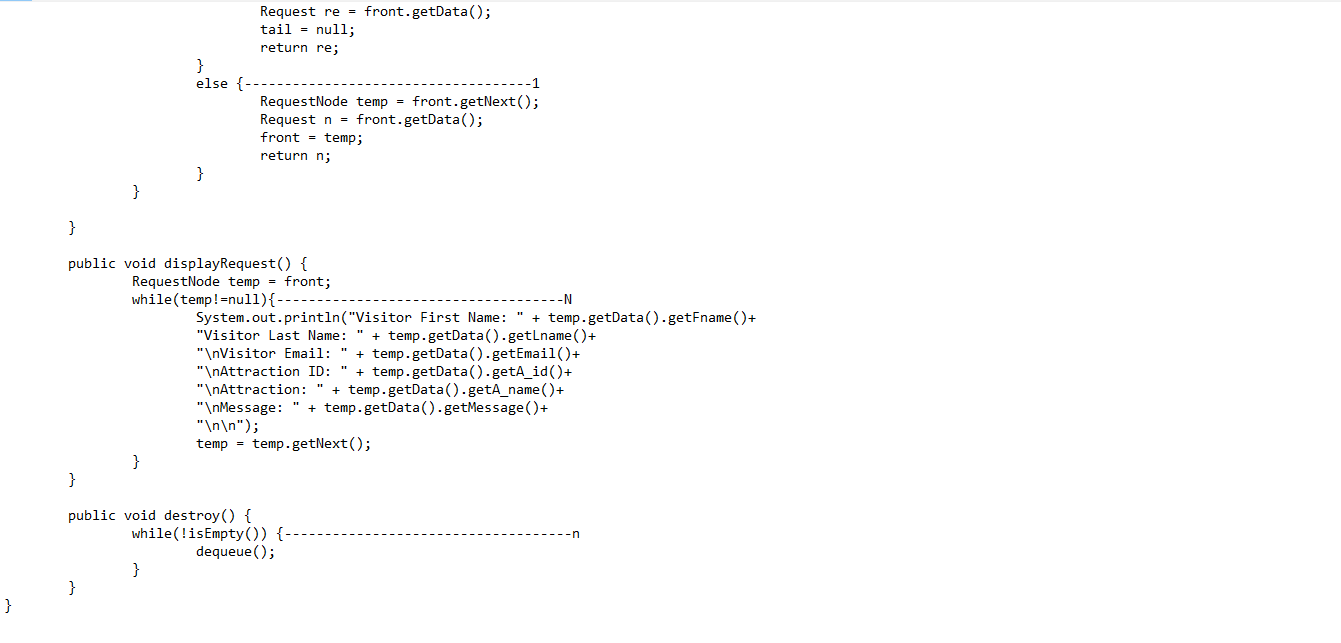
f(n)= 3+3+3+(1)+(1)+(1)+(1)+3+(1)+(1)+(1)+(1)+(1)+(1)

f(n)= 22

f(n)= 1

O(f(n))= 1, Constant

RequestQueue.java



Analysis of RequestQueue.java

f(n)=1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+n+n

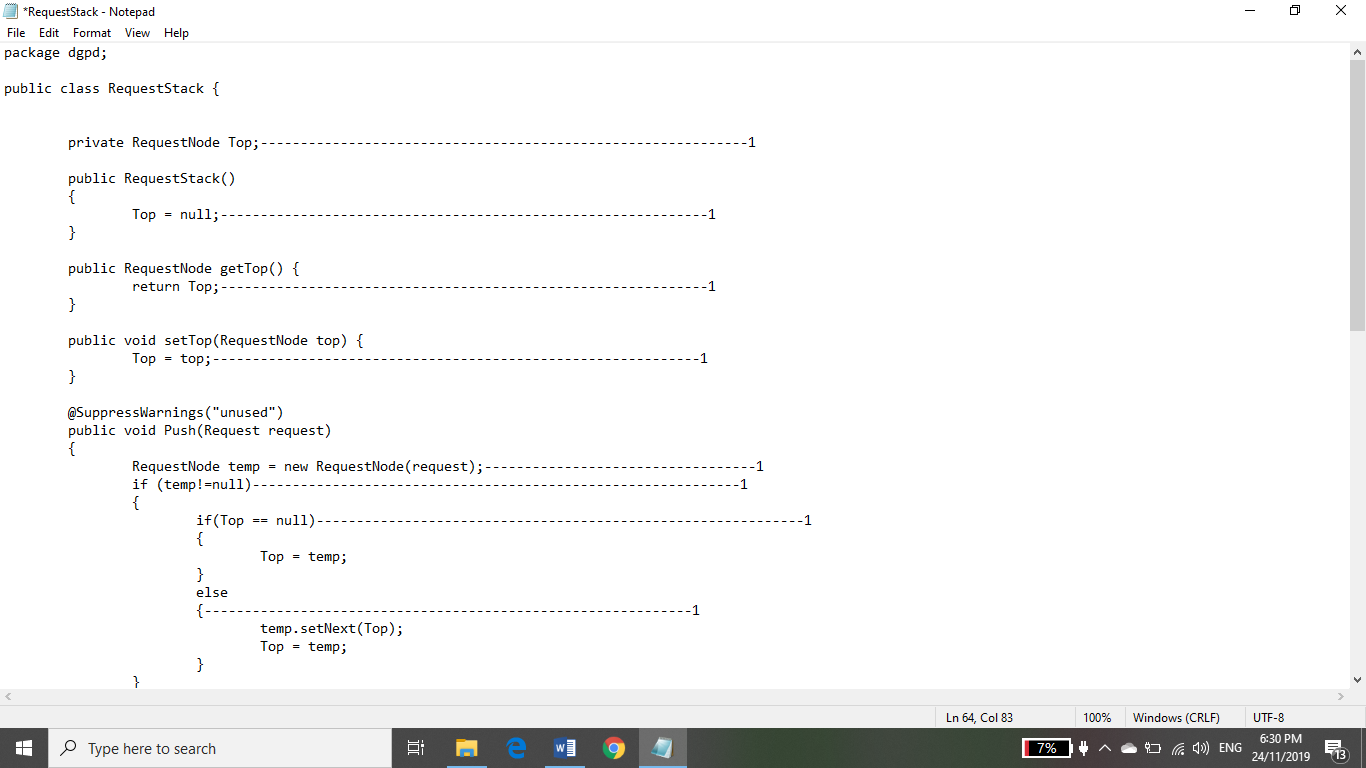
f(n)=17+(2n)

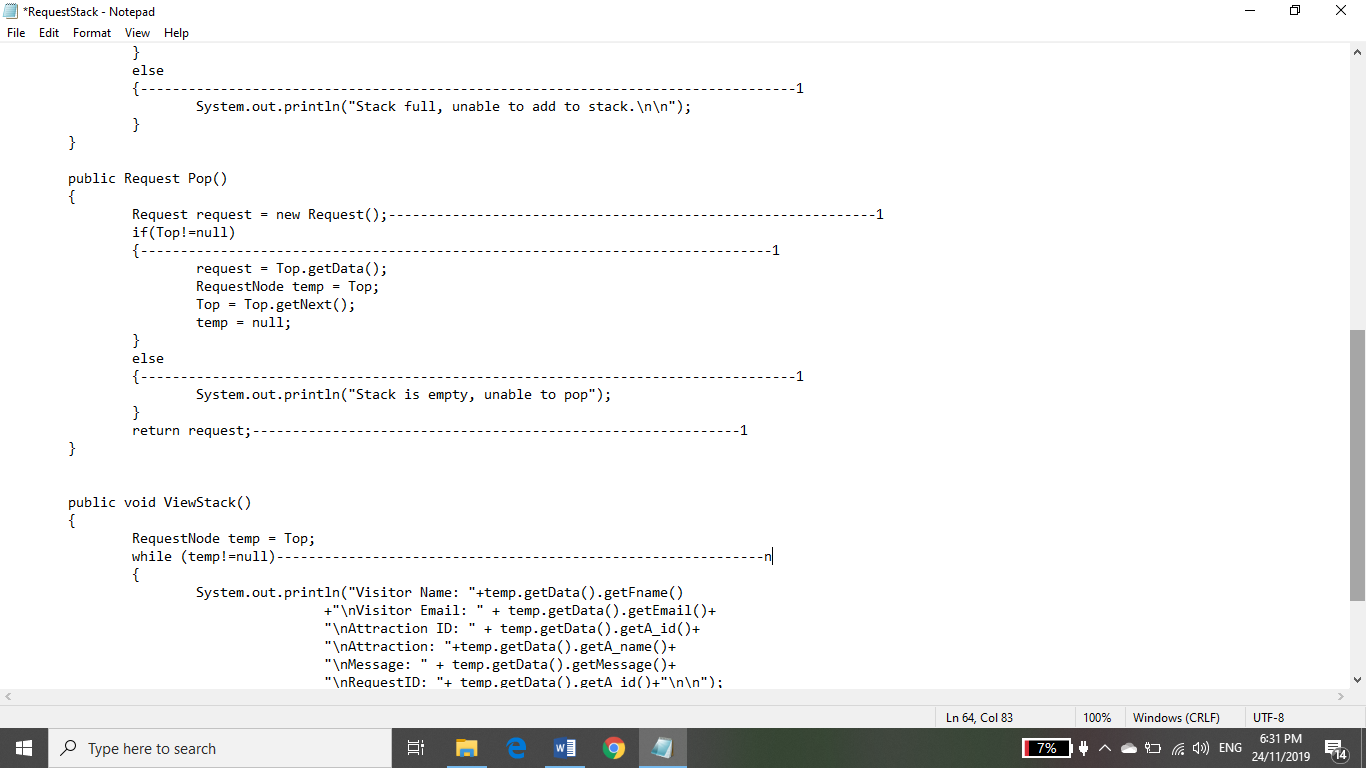
f(n)=2n

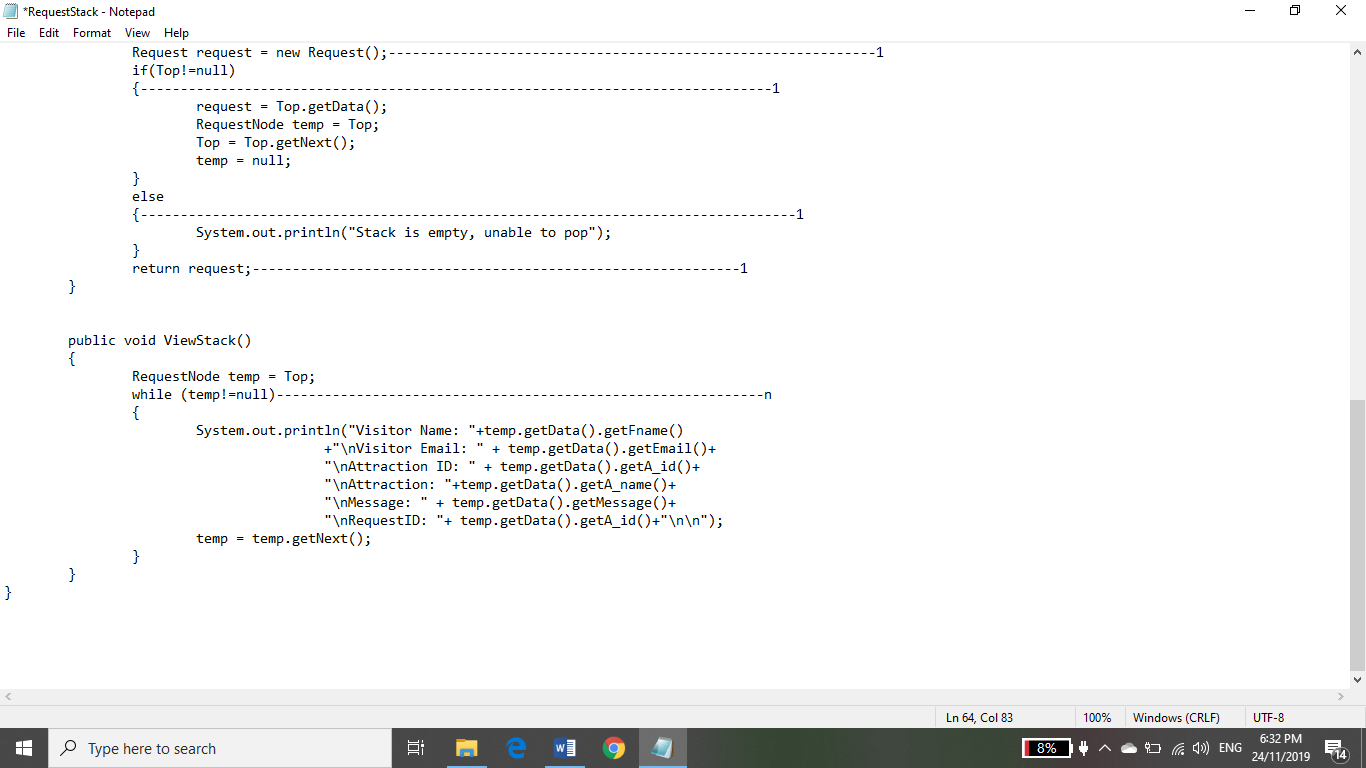
f(n)=n

O(f(n)= n, linear

## RequestStack.java







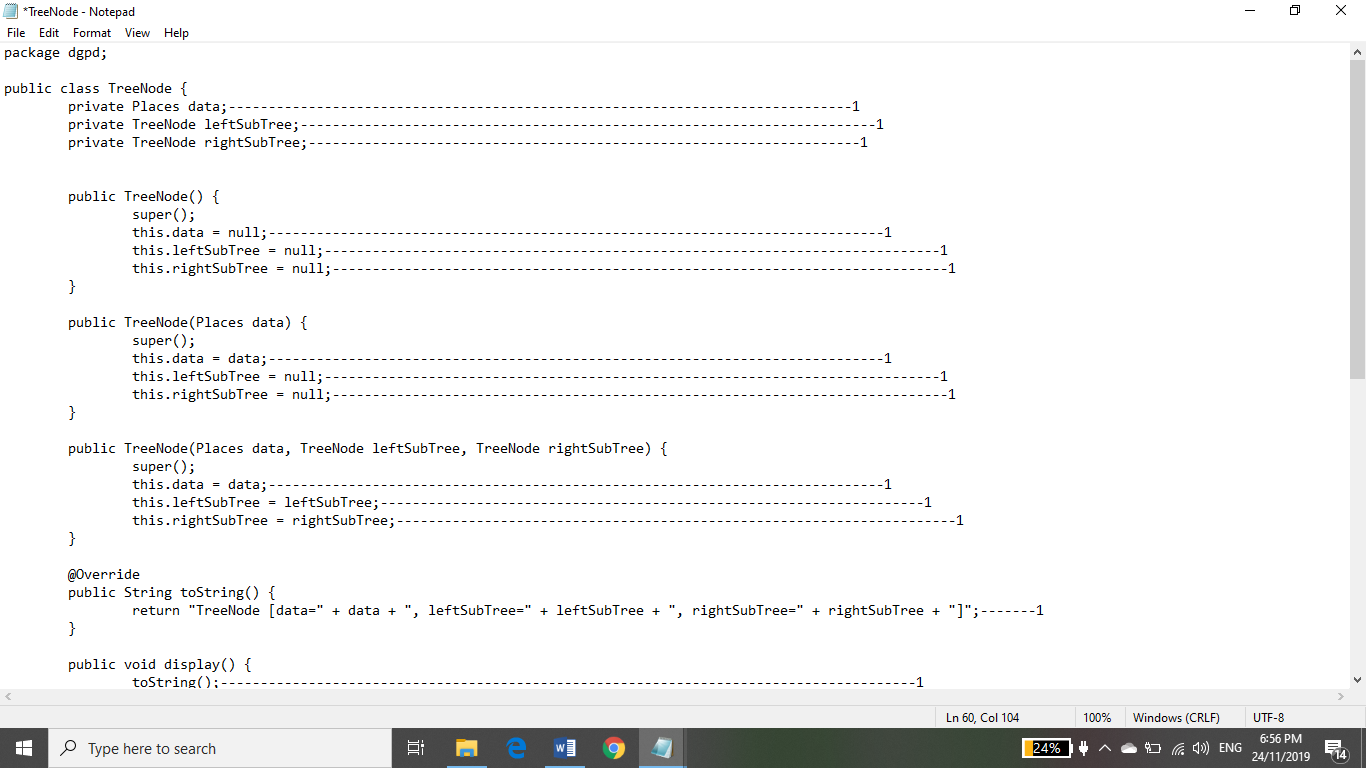
Analysis of RequestStack.java

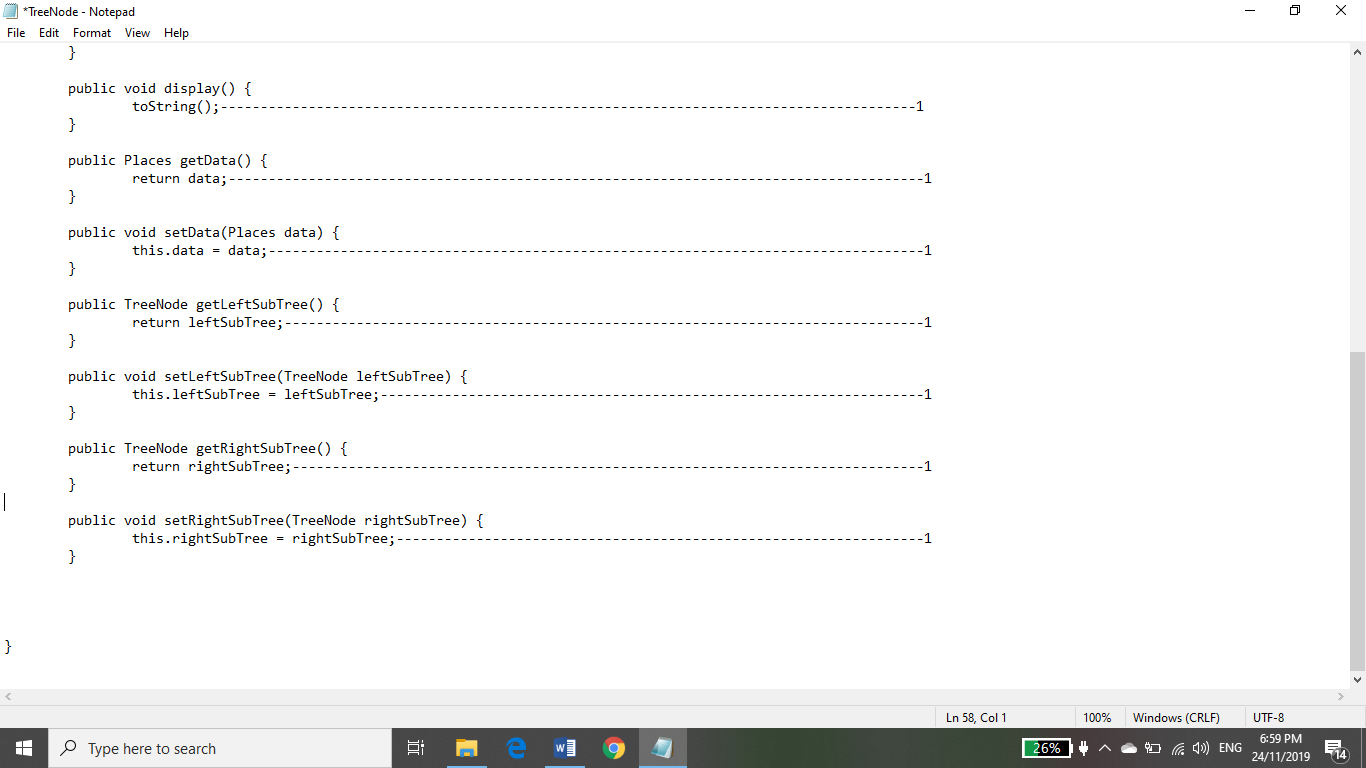
f(n) = 13 + n

f(n)= n

O(f(n)) = n, linear

## TreeNode.java





Analysis of TreeNode.java

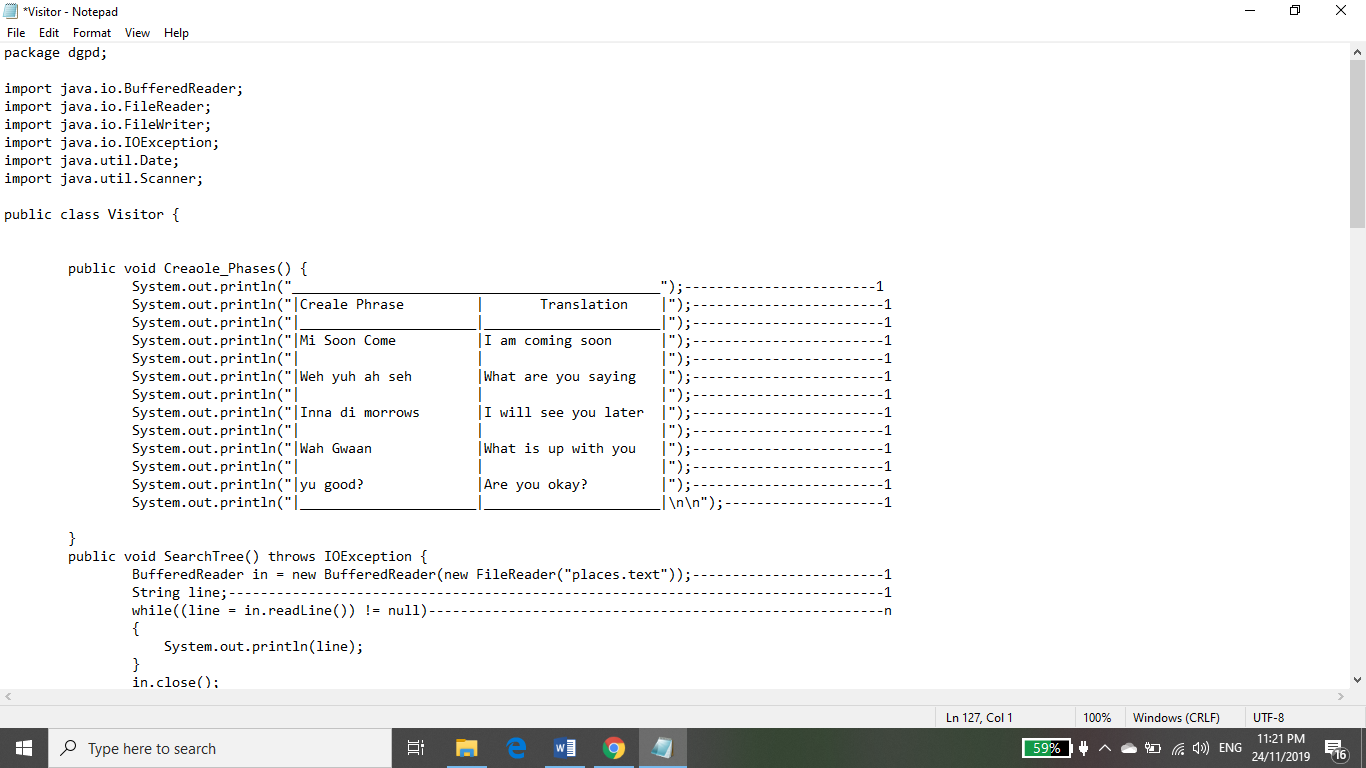
f(n) = 3+3+3+3+ 8

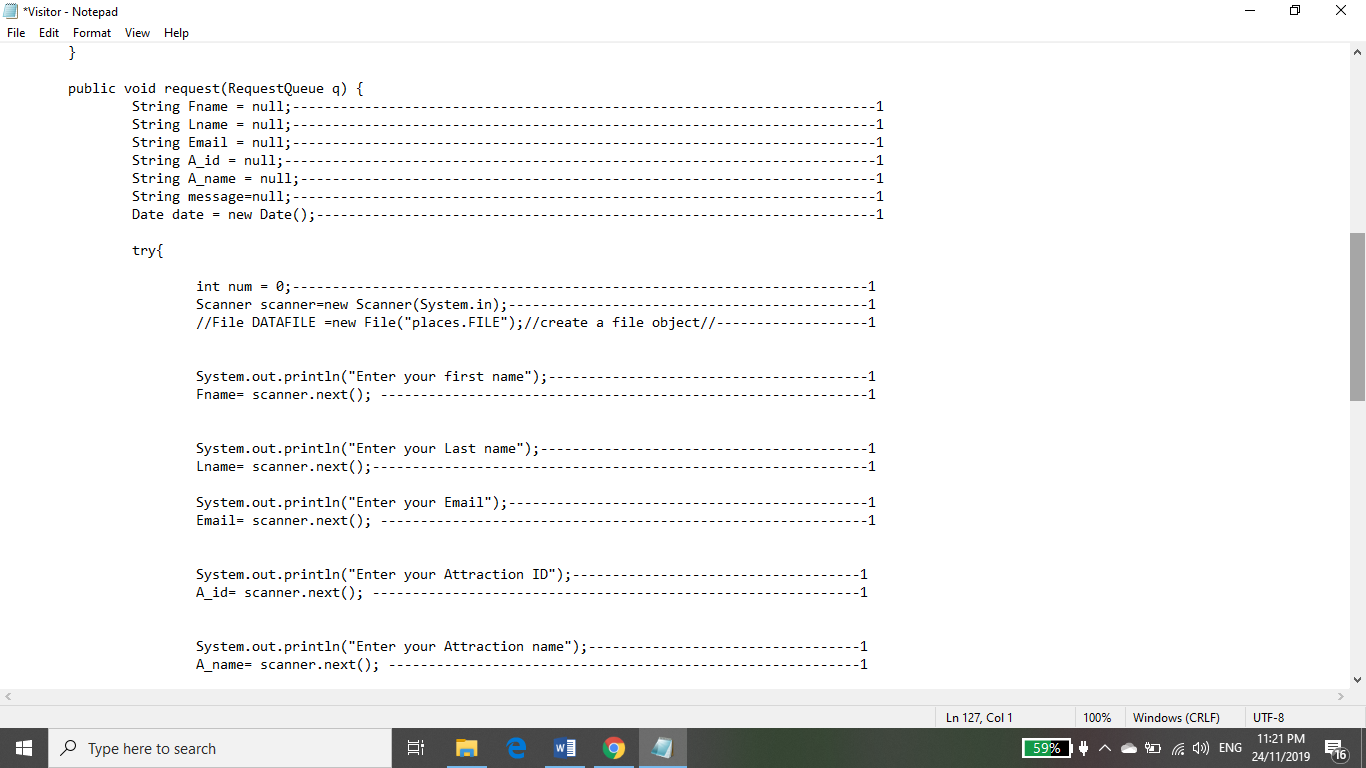
f(n)= 20

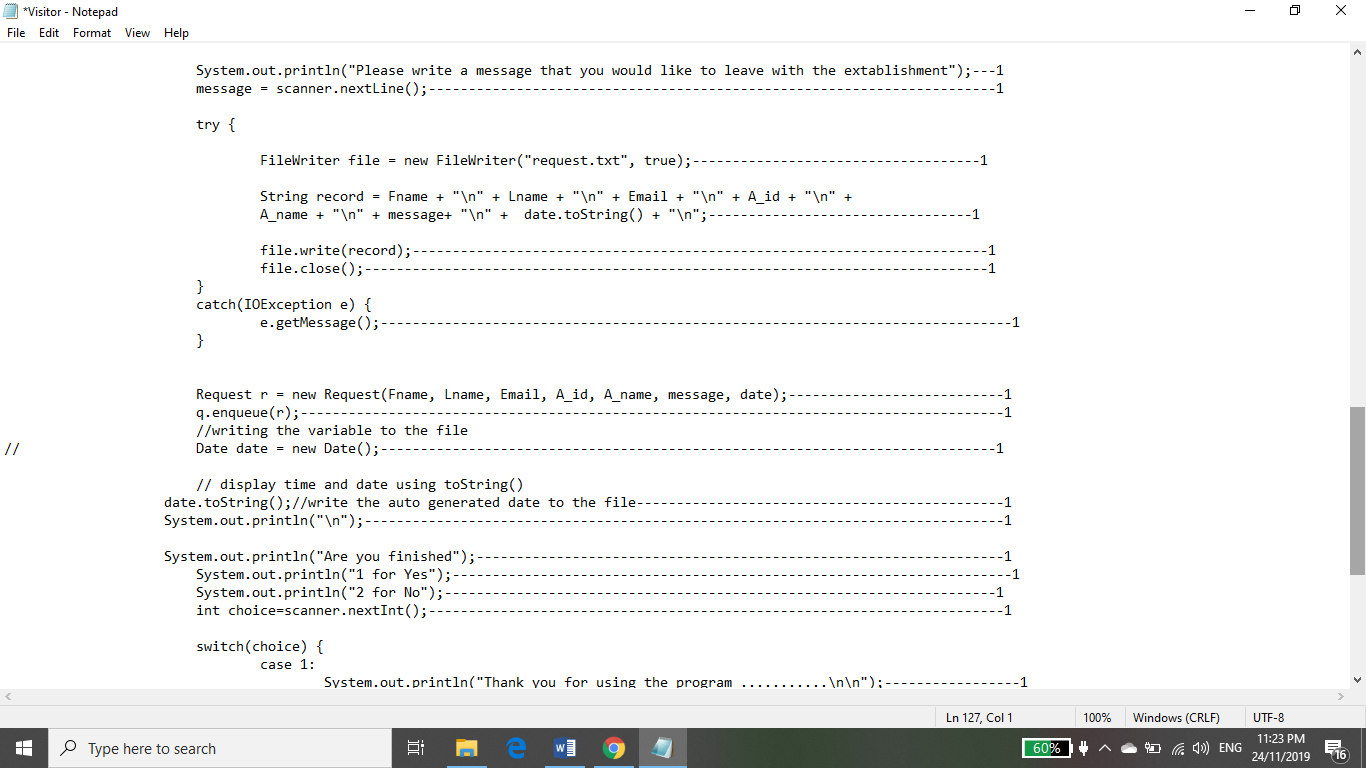
f(n)= 1

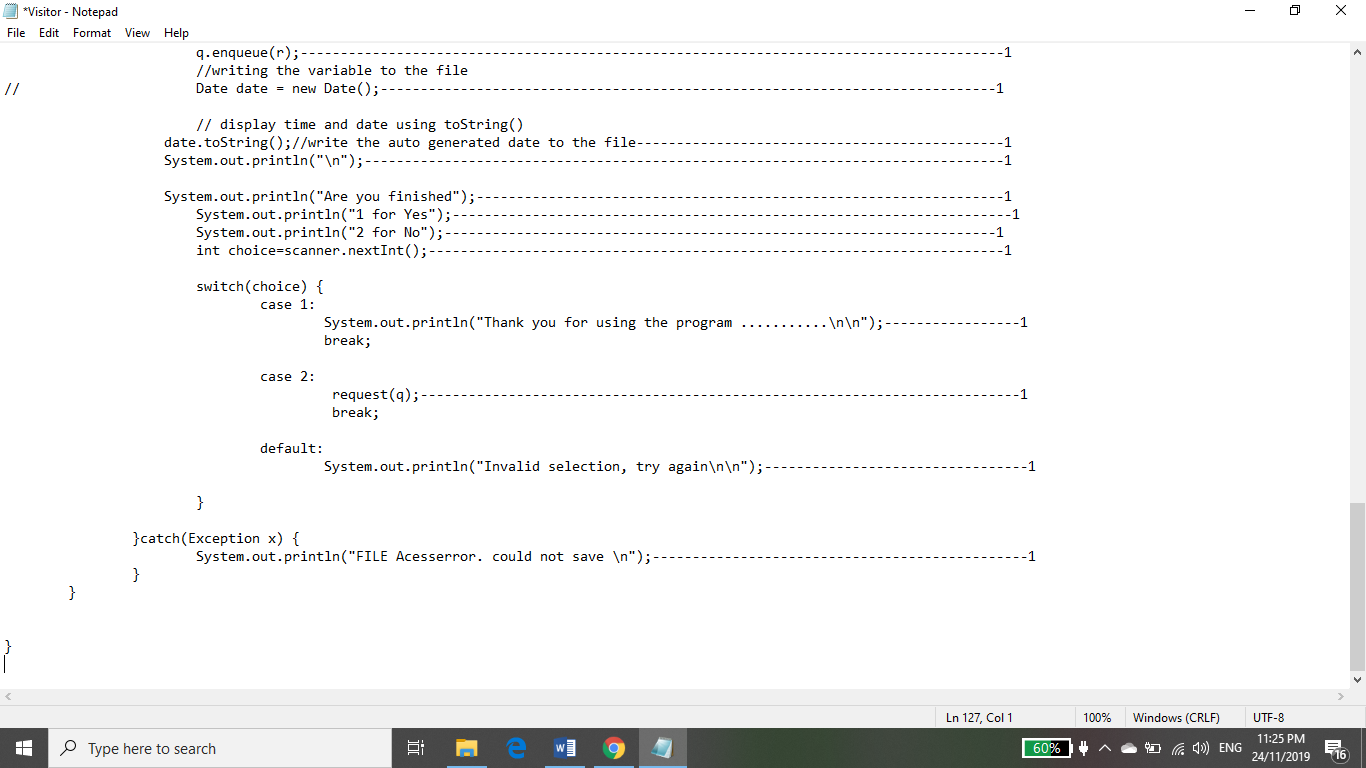
O(f(n)) = 1, Constant

## Visitor.java









Analysis of Visitor.java

f(n) = 13+(2)+ n+(7(15(4) +(1)+ 13

f(n)= 449 +n

f(n)= n

O(f(n)) = n, linear