# Java to Python Code Conversion Problems(convert Java Code to Python)

## Simple Data Types Conversion

public int factorial(int n) {  
 int result = 1;  
 for(int i = 1; i <= n; i++) {  
 result \*= i;  
 }  
 return result;  
 }

## Complex Data Structures

import java.util.HashMap;  
public char firstNonRepeated(String s) {  
 HashMap<Character, Integer> countMap = new HashMap<>();  
 for(char c : s.toCharArray()) {  
 countMap.put(c, countMap.getOrDefault(c, 0) + 1);  
 }  
 for(char c : s.toCharArray()) {  
 if(countMap.get(c) == 1) {  
 return c;  
 }  
 }  
 return '\0';  
}

## Object-Oriented Paradigm

public class BankAccount {  
 private double balance;  
  
 public BankAccount(double initialBalance) {  
 this.balance = initialBalance;  
 }  
  
 public void deposit(double amount) {  
 balance += amount;  
 }  
  
 public boolean withdraw(double amount) {  
 if(amount > balance) {  
 return false;  
 }  
 balance -= amount;  
 return true;  
 }  
  
 public double getBalance() {  
 return balance;  
 }  
}

## Recursive Functions

public int fibonacci(int n) {  
 if(n <= 1) {  
 return n;  
 }  
 return fibonacci(n - 1) + fibonacci(n - 2);  
}

## Functions with Varied Parameters

public int findMax(int... numbers) {  
 int max = Integer.MIN\_VALUE;  
 for(int num : numbers) {  
 if(num > max) {  
 max = num;  
 }  
 }  
 return max;  
}

## Exception Handling

import java.io.File;  
import java.io.FileNotFoundException;  
import java.util.Scanner;  
  
public String readFile(String filename) {  
 try {  
 File file = new File(filename);  
 Scanner scanner = new Scanner(file);  
 StringBuilder content = new StringBuilder();  
 while(scanner.hasNextLine()) {  
 content.append(scanner.nextLine()).append("\n");  
 }  
 scanner.close();  
 return content.toString();  
 } catch(FileNotFoundException e) {  
 return "File not found!";  
 }  
}

## Java Libraries and Dependencies

import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
  
public Date parseDate(String dateStr) {  
 SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");  
 try {  
 return format.parse(dateStr);  
 } catch(ParseException e) {  
 return null;  
 }  
}

## Whole Project Conversion

public class Book {  
 private String title;  
 private String author;  
  
 public Book(String title, String author) {  
 this.title = title;  
 this.author = author;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public String getAuthor() {  
 return author;  
 }  
}  
  
public class Library {  
 private List<Book> books = new ArrayList<>();  
  
 public void addBook(Book book) {  
 books.add(book);  
 }  
  
 public List<Book> getBooks() {  
 return books;  
 }  
}

## Functional Paradigm

import java.util.List;  
import java.util.stream.Collectors;  
  
public List<Integer> filterAndSquare(List<Integer> numbers) {  
 return numbers.stream()  
 .filter(n -> n % 2 == 0)  
 .map(n -> n \* n)  
 .collect(Collectors.toList());  
}