1. Fill in the missing code for implementing an Angular service named **DataService** which fetches array of **Employee** objects in JSON from the URL [http://web422.ca/api/employees](http://web422.ca/api/photos). Assume that all required imports are in place and the **Employee** class is present. [4 marks\]

**… …**

import { Employee } from "./employee";

@Injectable({ providedIn: 'root' })

export class DataService {

  constructor(  ) { } //1 private http: HttpClient

  getEmployees(): > { // 1 Employee

; //2

  }

}

Observable**<**Employee[]**>** {

**return** **this**.http.**get<**Employee[]**>**('[http://web422.ca/api/employees](http://web422.ca/api/photos)')

}

2. Fill in the missing code in the following **EmployeesComponent** which uses the **DataService** to get the data of **Employee** array by calling the service’s **getEmployees**() method and stores them in the property **employees** (see below). Assume that all required imports and the @Component decorator are in place, and the **Employee** class is present. [4 marks\]

**… …**

import { Employee } from "./employee";

export class EmployeesComponent implements OnInit {

  employees: Employee[\];

  sub: any;

  constructor(  ) { } //1 private data: DataService

  ngOnInit() {

; //2 this.sub=this.data.getEmployees()

 }

  ngOnDestroy() {

 // 1 **if**(**this**.sub){**this**.sub.unsubscribe();}

  }

}

(a) Fill in the blanks for the following code (file: "src/app/app-routing.module.ts") to enable routing in an Angular application, according to the following specification: [6 marks\]

* The "home" route, ie: "localhost:4200/home" displays the HomeComponent defined in "home.component.ts"
* The "students" route, ie "localhost:4200/students" displays the StudentsComponent defined in "students.component.ts"
* The "student" route with the parameter of **id**, e.g. "localhost:4200/student/006" displays the StudentComponent (wih the id of "002") defined in "student.component.ts"
* When a page (route) is not found, the PageNotFoundComponent defined in "pageNotFound.component.ts" is displayed
* The default route, ie: "localhost:4200" will redirect to the "/home" route (using the method discussed in class)

**File: src/app/app-routing.module.ts**

import { NgModule } from '@angular/core';

… …

// Define constant routes with value of array of route objects

 = [ **const** routes: Routes **= [**

 , {path: 'home', component: HomeComponent}

 , {path: 'students', component: StudentsComponent}

 , {path: 'student/:id', component: StudentComponent}

 , {path: '\*\*', component: PageNotFoundComponent}

 {path: '', redirectTo: '/home', pathMatch: 'full'}

\];

@NgModule({

  imports: [RouterModule.forRoot(routes)\],

  exports: [RouterModule\]

})

export class AppRoutingModule { }

(b) Fill in the blank to allow the link to be used to navigate to the Students component within the Angular (SPA) app: [1 mark\] 

<a >Students</a>  routerLink="/students"

(c) Fill in the blanks for the following StudentsComponent according to the following specification: [3 marks\]

* When the component is initialized, the "pageNum" value is set by accessing the value of the "page" query parameter used to display this component

import { Component, OnInit } from '@angular/core';

import ; { StudentsComponent} from './StudentsComponent'

@Component({

  selector: 'app-students',

  templateUrl: './students.component.html',

  styleUrls: ['./students.component.css'\]

})

export class StudentsComponent implements OnInit {

  pageNum: number;

  constructor() { } private router: Router

  ngOnInit(): void {

 ; this.router.navigate(['/students'], { queryParams: { page: pageNum } });

  }

}

**Question 3**

Fill in the blanks based on the “**TODO**” instructions to complete the code for Angular component interaction:

**File 1: app.component.ts**

import { Component, OnInit } from '@angular/core';

@Component({

  selector: 'app-app',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css'\]

})

export class AppComponent implements OnInit {

  appMsg: string;

  constructor() {}

  ngOnInit(): void {}

// **TODO:** implement the getMessage function and assign the passed in message to the value of the appMsg property. [1.5 marks\]

 { getMessage

 ; return appMsg

  }

}

**File 2: app.component.html**

//  **TODO:**  Render the kid component (below, as child component)  and pass event (with message) to the App component and handle the event with the getMessage method. [2.5 marks\]   



**File 3: kid.component.ts**

import { Component, OnInit, Output, EventEmitter } from '@angular/core';

@Component({

  selector: 'app-kid',

 templateUrl: './kid.component.html',

  styleUrls: ['./kid.component.css'\]

})

export class KidComponent implements OnInit {

 // **TODO:** Create a property named “**kidEvent**” which is able to emit messages. [1 mark\]

 ; kidEvent: String

  constructor() { }

  ngOnInit(): void { }

  emitEvent(){

   // **TODO:** implement the function to emit message "Hello, AppComponent" to App component. [1 mark\]

 ;  return "Hello, AppComponent"

  }