Typescript and Angular QB

1. what is the output of the below given code snippet?

```
let name: string = `  HSBC     `;

let year: number = 5;

let sentence: string = `Hello, I work in ${ name }.

I'll be completing ${ year + 1} years next month.`;

console.log(sentence)
```

a) Hello, I work in HSBC .
   I'll be completing 6 years next month.

b) Hello, I work in name.
   I'll be completing year + 1 years next month.

c) Hello, I work in ${ name }.
   I'll be completing ${ year + 1} years next month.

d) Hello, I work in undefined.
   I'll be completing NAN years next month.

2 Which is/are the correct Expression/s to define the following function as Arrow function?

```
var calculateInterest = function (amount, interestRate, duration) {

return amount * interestRate * duration / 12;

}
```

1. var calculateInterest = (amount, interestRate, duration) => amount * interestRate * duration / 12;

2. var calculateInterest = (amount, interestRate, duration) => {

return amount * interestRate * duration / 12;}

3. var calculateInterest = function (amount, interestRat, duration) => amount * interestRate * duration / 12;

a) Both 1 and 2.
b) All of these.
c) Both 1 and 3.
d) Only 2.

3 What is the output of below code:

```
1.  var addition = (x:number)=> x=x+10;
2.
3.  var display = (x)=>{
4.                  if(typeof x=="number"){
5.                  addition(x);
6.                  console.log(x);
7.                  console.log(x+" is numeric")
8.          } else if(typeof x=="string"){
9.                  console.log(x+" is a string")
10.         }
11.}
12.
13.
14. //function Call
15.
16. display("Jason");
17. display(123);
```

a) Jason is a string
   123 is numeric
   133
b) Jason is a string
   133
   123 is numeric
c) 133 is numeric
   133
   Jason is a string
d) Jason is a string
   123
   123 is numeric
e) Not Answered

4 Predict the output of the below code.

```
interface Product{
        productId?:number;
        productName:string;
}

function
getProductDetails(productobj:Product={productName:'Mobile',productCategory:'Gadge
t'})
:string
{
                return "The product name is "+productobj.productName;

}
let productDetails:string=getProductDetails();
console.log(productDetails);
```

a) The product name is undefined
b) The product name is Mobile
c) Compilation Error because incorrect value is assigned to default parameter
d) None of these

5 Observe the code snippet and choose the right answer from the options provided

```
class University{
public universityCode : number;
protected courseCode: number;
protected status: string;
constructor(universityCode:number, courseCode:number){
                this.universityCode = universityCode;
                this.courseCode = courseCode
}
}
class College extends University{
private collegeId: number;
        constructor(universityCode:number, courseCode:number, collegeId:
number) {
                super(universityCode, courseCode);
                this.collegeId = collegeId;
        }
        getCollegeDetails():void{
```

```
                    console.log(this.collegeId+" offers the course
        "+this.courseCode);
                }
        }
    var college: College = new College(2001, 403, 23);
    console.log("University Code :"+college.universityCode)
    college.getCollegeDetails();
```

a) Compilation Error
b) University Code :undefined
23 offers the course 403
c) University Code :2001
23 offers the course 403
d) University Code :undefined
23 offers the course undefined

6 What is the output if the below code is executed

```
class Lathe{
        static latheCount : number = 100;
        static updateLatheCount():number{
                return(Lathe.latheCount++);
}
getLatheName():string{
        return("Lathe Name : MYSLATHE"+Lathe.updateLatheCount());
}
}
var lathe:Lathe = new Lathe();
console.log(lathe.getLatheName());
```

a) Error: static method cannot be accessed by non static method
b) Error: static variable once initialied cannot be changed
c) LatheName : MYSLATHE101
d) LatheName : MYSLATHE100

7 Consider the below code

```
class Book {
   title: string;
   price: number;
   author: string;

   constructor(title, price, author, public publisher?) {
      this.title = title;
      this.price = price;
      this.author = author;
   }

   display() {
      console.log(this.title, this.price, this.author, this.publisher)
   }

}

let book1 = new Book("Typescript Basics", 100, "ETA", "  HSBC     ");
book1.display();
let book2 = new Book("Angular Basics", 200, "ETA");
book2.display();
```

What will be the console output of the above code

a) error: this.publisher is not a instance varianle in display method
b) error: constructor cannot have a optional parameter
c) error: public parameter cannot be optional
d) Typescript Basics 100 ETA HSBC
Angular Basics 200 ETA undefined

8 Predict the output of the below code.

```
1.  interface Player
2.  {
3.    playerId:string;
```

```
4.          play();
5.      }
6.      class Team
7.      {
8.              teamId:string;
9.              playerId:string;  //captain id
10.             play()
11.             {
12.                     console.log('I am playing in a Team')
13.             }
14.     }
15.
16.     class GameClass extends Team implements Player
17.     {
18.
19.     }
20.
21.     new GameClass().play();
22.
```

a) Compilation Error at line-16 ‘play() method of Player is not implemented in GameClass’
b) Compilation Error at line-16 ‘Property playerId is missing in type GameClass’
c) Compilation Error at line-21 ‘play() method can not be found in type GameClass’
d) I am playing in a Team

9

Predict the output of the following code snippet used to validate the employee ID and employee Mail Id of the employees of Infosoft.

```
interface Employee{
        employeeId: number;
        employeeName: string;
        employeeMailId: string;
}
function validateEmployee(employee:Employee):string{
let empIdValidation : boolean = employee.employeeId > 100000 &&
 employee.employeeId <= 999999;
let emailIdValidation : boolean = employee.employeeMailId.split('@')[1]=="infosoft.com";
         if (empIdValidation && emailIdValidation)
                 var result: string = "Details of " + employee.employeeName + " validated
succesfully";
```

```
        else
            var result: string = "Validation of " + employee.employeeName + " unsuccesfull";
    return result;
}

let employee = {employeeId:343233, employeeName:"Arun Kumar", employeeMailId :
"arunk@infosoft.com", employeePhone : ”9767543357”};
console.log(validateEmployee(employee));
```

a) Details of Arun Kumar validated succesfully
b) Error: variable result is declared inside if/else block cannot be returned outside the block
c) Validation of Arun Kumar unsuccesfull
d) Error: employee object has more parameters than the interface

10 Predict the ouput

```
interface Customer{
    customerId : string;
    customerName: string;
}
interface RegularCustomer{
    retailCardNo : number;
    retailOutletId : string;
}
interface PrivilegeCustomer extends Customer, RegularCustomer{
    festiveBonus: number;
}
function fetchCreditLimit(privilegeCustomer:PrivilegeCustomer):number{
    if(privilegeCustomer.retailCardNo >= 1000 && privilegeCustomer.retailCardNo <= 5000)
return privilegeCustomer.festiveBonus*3;
    else
return privilegeCustomer.festiveBonus*2;

}
let customerOne : PrivilegeCustomer={
    customerId : "C13132",
    customerName : "Umesha",
    retailCardNo : 1230,
    retailOutletId : "R1001",
    festiveBonus : 2500
}
console.log("Festive Credit Limit :Rs."+fetchCreditLimit(customerOne)+"/-");
```

a) Festive Credit Limit :Rs.7500/-
b) Festive Credit Limit :Rs.5000/-

c) Multiple inheritence is not supported for interfaces

d) No Error, No output

*_____*_____*

1. The below Angular code snippet has some lines of code missing. Please identify the lines to be placed as per the requirement mentioned below.

    1. Disable the respective button if the person in male/female (Line 8, Line 11)

    2. On click of a button, the respective person's details should be visible (Line 8, Line 11)

```
1 @Component({
2       template:`
3               <div>
4       <h1 [textContent]='"user is" +person.name'> </h1>
5       <table>
6       <tr>
7       <td [attr.colspan]='1+1'>
8       <button >Female</button>
9       </td>
10      <td [attr.colspan]='1+1'>
11      <button>Male</button>
12      </td>
13      </tr>
14      <tr>
15      <td>{{person.rating}}</td>
16      <td>{{person.address}}</td>
17      </tr>
18      </table>
```

```
19        </div>
20 `
21 })
export class AppComponent{
private female={
   name:'Jenny Wesley' ,
   gender:'F',
   rating:4,
   address:'Wesley states'
   }
 private male={
   name:'Ross Green',
   gender:'M',
   address:'New York'
   }
   private person=this.female;
}
```

a) Line 8: <button (click)='person=female' [disable]='person.gender=="F"'>
Line 11: <button (click)='person=male' [disable]='person.gender=="M"'>

b) Line 8: <button (click)='person=female' [disabled]='person.gender=="F"'>
Line 11: <button (click)='person=male' [disabled]='person.gender=="M"'>

c) Line 8: <button (click)='person=female' [disabled]='person.gender=="F"'>
Line 11: <button (click)='person=male' [disabled]='person.gender=="M"'>

d) Line 8: <button (click)='person=female' [disabled]='person.gender=="F"'>
Line 11: <button (click)='person=male' [disabled]='person.gender=="M"'>

2. In the below given code snippet, what is the correct statement to be placed at line number 7 to bootstrap an AppComponent class in Angular?

```
1  import { NgModule } from '@angular/core';
2  import { Browsermodule } from '@angular/platform-browser';
3  import { AppComponent } from './app.component';
4  @NgModule({
5    imports: [BrowserModule],
6    declarations: [AppComponent],
7    ....
8  })
9  export class AppModule { }
```

a. bootstrap: []
b. bootstrap: [AppComponent]
c. bootstrap: [app.component]
d. bootstrap: AppComponent

3. Which of the following is/are the correct way to bind a click event to the button in Angular?

i) <button (click) = "onSubmit(userName.value, password.value)">Login</button>

ii) <button on-click = "onSubmit(userName.value, password.value)">Login</button>

a. Only (i) is correct
b. Only (ii) is correct
c. Both (i) and (ii) are correct
d. Neither (i) nor (ii) are correct

4. Consider the below EmpService class in Angular with all the required code and imports. Select an appropriate statement to be written at Line 6 to complete the code for fetching the data from emp.json file.

```
1 @Injectable()
2 export class EmpService{
3 constructor(private http: HttpClient){}
4 private empUrl = 'emp.json';
5 getEmp(): Observable<Emp[]>{
```

```
6 ….
7 .pipe(tap(data => console.log(JSON.stringify(data))),
9   catchError(this.handleError));
10 }}
```

a) return this.ajax(empUrl)
b) return this.http.get(emp.json)
c) return this.http.get<Emp[]>(this.empUrl)
d) return Promise.resolve(empUrl)

5. Consider the following code snippet in Angular. What is the purpose of observe option used in line number 11?

```
1 ...
2
3 @Injectable()
4 export class BookService {
5
6  private booksUrl = './assets/books.json';
7
8  constructor(private http: HttpClient) { }
9
10   getBooks(): Observable<HttpResponse<Book[]>> {
11     return this.http.get<Book[]>(this.booksUrl, { observe: 'response' }).pipe(
12       tap(books => console.log(books.headers.get('Date'))),
13       catchError(this.handleError))
14   }
15  ...
16 }
```

a) Observes the response coming in from the server

b) Returns the data in different formats
c) Fetches the full response which includes headers and body
d) Retries the request if server is not responding

6. Akaash wants to develop a form in model driven approach in Angular. Which is the necessary module that needs to be added?
    a. imports: [FormsModule]
    b. imports: [ReactiveFormsModule]
    c. declarations: [FormsModule]
    d. declarations: [ReactiveFormsModule]

7. Sam wants to use a service class called 'BookService' in a 'BookComponent' class in Angular. Which of the following statement he should use to inject BookService class into BookComponent?
    a. constructor(private bookService: BookService)
    b. constructor(bookService: BookService)
    c. we can directly use BookService class in BookComponent
    d. constructor(bookService: new BookService())

8. Consider the below code snippet in Angular. Consider all imports are done and route.module.ts file contains the path definition. Through routing we should navigate from app.component.html to about.component.html. Then from about.component.html we should navigate to register.component.html. What will be the correct code to be placed at line number 1 in about.component.html to navigate from about.component.html to register.component.html page.

route.module.ts

```
const routes: Routes  = [

 { path: 'register', component: RegisterComponent },

 { path: 'about', component: AboutComponent }

];
```

about.component.html

Angular is the most popular course in current trend. To register yourself please click the below link.

//Line 1

a) <a [href]="['/register']">Register
b) <a [router]="['/register']">Register
c) <a [routerLink]="['/register']">Register
d) Register

9. Judge the result of the Angular code given below.

```
import { Routes, RouterModule } from '@angular/router';

import { WelcomeComponent } from './welcome/welcome.component';

import { LoginComponent } from './login/login.component';


const routes: Routes = [

  { path: '', redirectTo: 'welcome', pathMatch: 'full' },

  { path: 'welcome', component: WelcomeComponent },

  { path: 'login', component: LoginComponent }

];


export const routing = RouterModule.forRoot(routes);
```

a) Compiles successfully
b) Compiles and successfully runs
c) Compiles successfully but runtime error occurs.
d) Results in compilation error

10. Sam wants to deploy his Angular application with all the optimization features included like AOT compilation, production mode, bundling, minification, uglification and dead code elimination. Which of the following command he should use which includes all of these optimization features by default?
    a. ng build
    b. ng serve
    c. ng build --aot
    d. ng build --prod

11. Wilson and his team has created an angular Application for Furniture Shopping. They had created different modules named Login, NewArrivals, GetAllFurnitures, MyPurchase and HomeDelivery. As the app takes little longer time to load every module they decided to load MyPurchase, HomeDelivery modules only when the user wants to view. Suggest them a solution to make it easier.
    a. use @LoadChidren decorator at MyPurchaseComponent and HomeDeliveryComponent
    b. use loadChildren property for MyPurchase and HomeDelivery paths given in Routes module
    c. use link for MyPurchase and HomeDelivery in GetAllFurnitures page for navigate only when user clicks the link
    d. no option is there in angular to do such loading

12. In Angular rouitng, consider the base url as 'http://angularLibrary.com'. Choose the updated Url when the below routes are encountered:

```
[
{path: 'books', component: BooksComponent},
{path: 'book/:id', component: BookDetailComponent},
{path: '**', component: PageNotFoundComp
]
```

    a) http://angularLibrary.com/books
       http://angularLibrary.com/book/6
       http://angularLibrary.com/toCart

b) http://angularLibrary.com/books

http://angularLibrary.com/book:6

http://angularLibrary.com/toCart

c) http://angularLibrary.com/BooksComponent

http://angularLibrary.com/book/6

http://angularLibrary.com/toCart

d) http://angularLibrary.com/books

http://angularLibrary.com/books/6

http://angularLibrary.com/PageNotFound

13. In Angular, which of the following code snippet(s) is the proper way of creating a component with an injected service class?

Code 1:

```
import { Component, OnInit } from '@angular/core';
import { LoginService } from './login.service';
@Component({
 selector: 'app-login',
 templateUrl: './login.component.html',
 styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {
```

```
  constructor(private loginService: LoginService) { }
}
```

Code 2:

```
import { Component, OnInit } from '@angular/core';
import { LoginService } from './login.service';
@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css'],
  providers: [LoginService]
})
export class LoginComponent implements OnInit {
  constructor(private loginService: LoginService) { }
}
```

a) Both the given code snippets are valid
b) Only Code 1 is valid
c) Only Code 2 is valid
d) Neither of the given code snippets are valid

14. In Angular, what are the different ways to pass data from one component to other component from the options listed below?
    1. Parent child relation
    2. Routing parameters
    3. Services

a) All the given options are valid
b) Only (1) and (2) are valid
c) Only (1) is valid
d) None of the given options are valid

15. Sam wants to create a custom pipe in Angular which should return the length of the given string. What is the correct statement Sam should write at Line number 4 to implement the same?

```
//length.pipe.ts
1 @Pipe({name: 'length' })
2 export class LengthPipe implements PipeTransform
3 {
4       …{
5               return value.length;
6       }
7       }
```

a) transform(value: string)
b) LengthPipe(value: string)
c) length(value: string)
d) PipeTransform(value:string)