1)Angular command line interface enables us to run ng commands in the terminal
--True

2)Who developed and maintains TypeScript?
--Microsoft

3)The html selector for the default angular component is:
--app-root

4)To generate a new component called Card, we use the cli command:
--ng generate component Card

5)Highlight the line that contains the optional parameter in the interface.
--tags?: string[];

6)Which of the following are default types built into TypeScript? Select all that apply.
--number
string
any
boolean

7)Angular and AngularJS are one and the same.
--False

8)Which of the following can now be used with variable declarations? Select all that apply.
-- static
private
public
protected

9)Given the following code, the function processContent has:
processContent (content: Content):void {
}
--No return value

10)What keyword do we add before our interface definition in order to make it accessible as a JavaScript module out side of the file?
--export

11)The ngOnInit function gets called before the component is done being created
--False

12)What are *ngIf and *ngFor?
--Structural Directives

13)Which of the following is a value we can track and use in our ngFor (select all that apply)?
--index of the item
odd items
first item

14)Directives are a superset of:
--component

15)Components are a subset of:
--Directives

16)What part of the component definition tells us the component's tag name?
--The selector

17)To generate a new component called Card, we use the cli command:
--ng generate component Card

18)If we have an ngFor on a content-card component set to iterate over 5 Content items, what html will be outputted on the page?
--5 content-card blocks with a different content item in each

19)Which of the following will correctly use ngFor?
--<app-content-card *ngFor="let myItem of listOfContent" [content]="myItem"></ app-content-card>

20)Angular command line interface enables us to run ng commands in the terminal
--True

21)Using the double curly brace syntax is a our default form of:
--Interpolation

22)Pipes cannot be applied to static string values
--False

23)The value output by using the double brace syntax is of the type:
--string

24)You cannot chain pipes
--False

25)Two way binding is represented by using:
--Square and round brackets around the expression

26)Performing array mutation will trigger pipes to re-evaluate arrays each time an element is added.
--False

27)The core function of pipes is to:
--Transform data

28)What is something you still can't do in a template statement?
--Use shorthand incrementors and decrementors such as ++ and --

29)With two way data binding, we no longer have to wait on certain page events to occur for the value on the page t
o update
--True

30)Some pipes that pre-exist in Angular include (select all that apply)
--Date
TitlecCase

31)Select all the following that are valid directive types:
--Structural
Template associated
Attribute

32)What object type gives us access to the properties of the tag our directive is acting on?

--ElementRef

33)What would be an example of a structural directive?
--*ngFor="let item of content"

34)We can pass input variables to our attribute directives via properties defined on the tag we're operating on.
--True

35)We can define private variables in a class constructor and use that variable throughout the class.
--True

36)Which of the following are valid event types? (select all that apply)
--blur
mouseenter

37)In order to manipulate the element in our attribute directive, what object type must we include in our constructor?

--ElementRef

38)In order to manipulate an element in a directive, what must we import into the directive?
--ELementRef

39)Which of the following are event types we've talked about in class? (Select all that apply)
--blur
click
mouseenter

40)*ngIf hides the element on the page if it evaluates to false
--False

41)Who developed and maintains Angular?
--Google

42)TypeScript is a superset of (select all that apply):
--ES2015
ES5
Javascript

43)TypeScript interfaces turn into object prototypes when they're compiled down to ES5 code.
--False

44)How do you set a variable to only be accessible within the class and its child classes, but not explicitly accessible otherwise?
--Declare it to be protected

45)To run commands like generate in the console, we add scripts to the package.json file, and need to run a special command to make them work. Fill in the blank below:
npm _________ generate pipe SafeHtml
--run-script

46)The square brackets used on html properties is an example of:
--Data Binding

47)Given the following code, what would the missing html be for declaring a card component tag?
Typescript:

```
import { Component, Input } from '@angular/core';
@Component({
   selector: 'app-card',
   templateUrl: './card.component.html',
   styleUrls: ['./card.component.scss']
})
export class CardComponent {
   @Input() item: string;
}
```

HTML:

```
<[A] [[B]]="currentItem"><[C]>
```

--app-card

48)In order for our component to accept a parameter in its html tag, what do we change in our component TypeScript file (select all that apply)?
--Add Input to our list of imports from angular/core
Declare a variable in out component class using the @Input() decorator

49)It a tag's *ngIf is false, none of the tag's html will be added to the page
--True

50)the @Component in our typescript file is a:
--Decorator

51) What does the following partial definition of a pipe output if the Content type passed in is null and dType is set to "Stories"?

```
export class DefaultTypePipe implements PipeTransform {
transform(content: Content, dType: string ): string {
return content.type || dType || "News";
}
}
```

--Stories

52)There are multiple ways to write each type of data binding
--True

53)The primary reason we import the PipeTransform interface is:
--To make use of its transform method defined by it

54)Which of the following is not a type of data binding?
--Component data binding

55)What is the name of the type of variable being used in this example?

```
<input #name>: {{ name.value }}
```

--Template reference variable

56)Template statements can be spotted because of:
--The round brackets around the event name

57)A directive is a component but a component is not a directive.
--False

58)Structural and attribute directives fully manipulate the DOM by adding, removing and manipulating elements
--False

59)HostListeners are used to:
--Watch an event trigger

60)Which of the following are structural directives we've discussed in class?
--*ngIf
*ngSwitch
*ngFor

61)Directives can define (select all that apply):
--A custom element
A way to manipulate an element on the DOM
A way to manipulate attributes of an element
A way to monitor events on elements and trigger functionality

62)In the provided image, select the keyword that defines what kind of event we are watching for
--click