

This exam has 25 problems on 9 pages. If the answer to a question is undefined or an empty string please write that in the answer box. If the code would result in an error write error. All problems are worth 5 points. Four class of problems: C class problems,(these are the easiest), B Class problems (These are a little more difficult) A class problems (Answers these correctly requires mastery of the material). The final class of problems is A\* problems.(These are extra credit problems and often require that you combine knowledge form prior classes)

Name: \_\_\_\_\_

Score:

**Problem 1.** On my honor I have neither given nor received any aid this exam.

**Problem 2.** [C] Consider the following class, how do you get the value 42 from an instance of X?

```
class X{
  get Y() {return 42;}
}
```

x.y | OK I added x = newX()

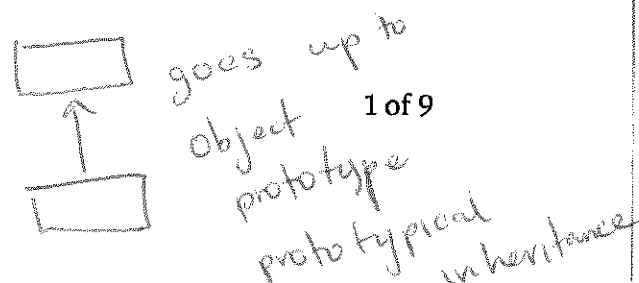
*capital*

*any instantiation is O.K*

**Problem 3.** [B] When the code below is run what will be console logged?

```
function logger(){
  console.log("executed function")
}
console.log(logger.prototype)
```

- executed function
- undefined
- an error message
- an object with a constructor property



*Extra credit*

**Problem 4. [A\*]** Why is it usually better to work with Object instead of Array to store a collection of records

- Most operating involve looking up a record and objects can do that better than arrays
- Adding a record to an object is significantly faster than pushing a record into an array
- Working with objects makes code more readable
- Objects are more efficient in terms of storage.

**Problem 5. [C]** What is the value of a.length

```
var a = ['wahoo', 'tribe', 'bulldogs']  
a[100] = 'alpha'
```

- 101 *← remember zero indexed*
- 3
- 4
- 100
- error

**Problem 6. [C]** Which of the following is an implicit parameter for every function in JavaScript

- argumentList
- args
- argsArray
- arguments
- none of the above

**Problem 7. [C]** What is the result of the following code?

```
var y = 1
var x1 = () =>{
  console.log(y)
}
```

```
var x2 = function(){
  var y = 2
  x1()
}
x2()
```

- 2
- 1
- undefined
- error

← declares a new variable  
y would have been

2 if y = 2 (without  
the var)

*Extra credit*  
**Problem 8. [A\*]** What is the name of the special JavaScript function whose execution can be suspended and resumed at a later point?

- Generator function
- Async/Await function
- Arrow function
- Promise function
- no such function exists.

**Problem 9. [A]** What does the following expression evaluate to ?

`[] == []`

- undefined
- FALSE
- []
- TRUE
- None of the above.

← compares two objects so  
it is looking at the  
memory location.

**Problem 10. [A]** What does the error: "TypeError: Cannot read property 'reduce' of undefined" mean?

- You are calling a method named reduce on object that does not exist
- You are are calling a method named reduce on an object that has a null value
- You are are calling a method named reduce on an object that's declared but has no value
- You are calling a method named reduce on an empty array.

**Problem 11. [B]** The problem below contains a syntax error, how would you fix it?

```
const person = (firstName, lastName)=>
({
  first: firstName,
  last: lastName
})
```

*make  
single  
line  
return* →

```
console.log(person("Jill", "Jack"))
```

- Add a return statement before the first curly brace.
- Replace the object with an array
- Wrap the object in parentheses.
- Call the function another file.
- None of the above.

**Problem 12. [C]** In which life-cycle method should you make request for data?

- ~~componentWillMount~~
- constructor
- ComponentWillReceiveProps
- ComponentDidMount
- None of the above

*correct answer* ↗

**Problem 13.** [B] Consider the JavaScript below what gets printed out.

```
obj = {  
  quad: function(func, x){  
    return func(this.x) - func(x)  
  },  
  x: 5  
}  
result = obj.quad(function mistrey(x){  
  return x % 2  
},2)  
  
console.log(result)
```



1

**Problem 14.** [B] Consider a layout with a two views. The view is a container view with a flex value of 1 and second view is a child view with dimensions of 100 x 200. Which styling snippet, if apply to first view, places the second view in the bottom right hand corner of the display. (These are just snippets so you can ignore syntax errors. Select all that apply.

- flex Direction: "column",  
justifyContent: "flex-end",  
alignImtes: "flex-end",
- flex Direction: "row",  
justifyContent: "flex-start",  
alignImtes: "flex-start",
- flex Direction: "column",  
justifyContent: "flex-start",  
alignImtes: "flex-end",
- flex Direction: "row",  
justifyContent: "flex-end",  
alignImtes: "flex-start",



Problem 15. [B] What does the following code log out?

```
for(var i = 0; i < 5; i++) {  
  setTimeout(function() {  
    console.log(i);  
  }, 1000);  
}
```

5, 5, 5, 5, 5

Problem 16. [C] Which of the following is/are true about setting the react native state. Shade the box if the answer is true

- Whenever setState() is called, the component also calls render() with the new state
- State updates can be merged by passing in an object to setState()
- Updating state directly is ideal when you want to re-render a component (using this.state.message="hi"; instead of this.setState({message: "hi"};)
- State updates can be asynchronous (i.e., setState() can accept a function with the previous state as its first argument)
- setState() should be called within component's render() method

Problem 17. [C] Consider the program below that uses ISomorphic fetch. Does the program contain any errors, if so on what line(s). If the program does not contain any errors, write \*\*no errors\*\*. Assume that line numbers start at top and begin with one.

```
1 async getData(){  
2   let response = fetch("http://validApi.edu")  
3   let parseObject = response.json()  
4   return parseObject  
5 }
```

2, 3 missing await response.json

is procceted asynchronous

**Problem 18.** [B] Consider the following promise chain. What gets printed out when the function runs.

```
let result = new Promise(function(resolve, reject) {
  setTimeout(() => resolve(10), 10000);
}).then(function(result) {
  return result/5;
})
result.then(result){
  console.log(result)
}
```

2

**Problem 19.** [C] How long does it take to print out the result in the code from the previous question. Specify your answer in seconds. It is OK to provide a range for example, less than second or between 10 seconds and 12 seconds.

10 to 12 seconds

**Problem 20.** [B] Consider the following program. What does it print out? If it prints an error write error.

```
const arr = [1,2,3,4,5];
let result = []
mapping = (c, i) => {
  return i % 3 == 0 ? result.push(c) : c
}
arr.map(mapping)
console.log(result.reduce((a, b) => b*b))
```

*0 1 2 0 1*  
*↑*  
*a is never used*

16 = 4x4

**Problem 21.** [C] Which on the following are advantages of the react native virtual DOM model.

- The developer does not have to track individual changes, the react native framework will calculate and apply only the difference to the DOM.
- Only parts of the real DOM/UI gets refreshed.
- The virtual DOM allows developer to render multiple versions of the DOM at the same time.
- The virtual DOM allows developers to clone the users DOM

**Problem 22.** [C] React Native requires that all components are pure functions which of the lines in the program below violate the pure function requirement. Assume that line numbers start at the top and begin with one.

```
1 export default function App() {  
2   constructor(props){  
3     super(props)  
4     props = {a: 12245}  
5   }  
}
```

↑ functional components  
must be pure fun

4

**Problem 23.** [C] Which of the following are advantages of building a React Native Application vs a completely native application. Shade in all answers that apply.

- Using React native allows developers to build Android and IOS applications using a single code base.
- React native applications provides developers with full access to all of the libraries and features supported by the android operating system.
- React native allows for remote debugging on both Android and IOS devices.
- React native allows for continuous updates without requiring users to go to the app store.

**Problem 24.** [C] There are two JSX elements next to each other on the display. Changing which of the following properties would result in increasing the distance between the elements. Shade in all that apply

- padding
- margin
- border
- alignment



**Problem 25. [B]** Consider following layout which of the css flex box snippets below moves the frogs to the correct location?

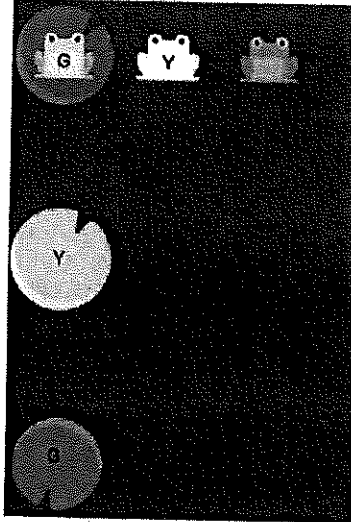


Figure 1: Layout Question 2

- ```
#pond {  
  display: flex;  
  flex-direction: column-reverse;  
  justify-content: space-between;  
}
```
- ```
#pond {  
  display: flex;  
  flex-direction: row-reverse;  
  justify-content: space-between;  
}
```
- ```
#pond {  
  display: flex;  
  flex-direction: column-reverse;  
  justify-content: space-evenly;  
}
```
- ```
#pond {  
  display: flex;  
  flex-direction: row-reverse;  
  justify-content: space-evenly;  
}
```

