This exam has 20 problems on 8 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

Name: _____   **Score:**

Problem 1. Consider the following snippet from a React Native Application. What is contained in the text JSX element during the first 7 seconds when screen is rendered.

```
constructor(props){
  this.state = {value: "1"}
}
componentWillMount(){
  setInterval( ()=>{
    this.setState({ value: "2"})
  }, 7000)
}
render(){
  return (
    <View style={styles.container}> <Text>{this.state.value - 2}</Text> </View>
  );
}
```

Problem 2. Consider the JavaScript below what gets printed out.

```
obj = {
    quad: function(func, x){
    return func(this.x) + func(x)
    },
    x: 3
}
result = obj.quad(function mistrey(x){
    return x % 2
    },2)

console.log(result)
```

Problem 3. Look at the following snippet, what is the code going to print out on line A?

```
function Person(first_name, last_name) {
    this.first_name = first_name;
    this.last_name = last_name;
}
Person.prototype.full_name = function(){
    return this.first_name + ' ' + this.last_name;
}
function Professional(title, first_name, last_name){
    Person.call(this, first_name, last_name);
    this.title = title;
}
Professional.prototype.prof_name = function(){
    return this.title + ' ' + this.first_name + ' ' + this.last_name;
}
var p = new Professional("Dr.", "Lisa", "Simpson");
console.log(p.prof_name()); //--- line A
console.log(p.full_name()); //--- line B
```

Problem 4. What gets printed out on Line B?

Problem 5. 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 6. What does the following code log out?

```
for(var i = 0; i < 10; i++) {
    setTimeout(function() {
      console.log(i);
    }, 10);
}
```

Problem 7. 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 8. Evaluate the following expressions, is it True or False for 1-5 and what is the result of 6. Shade the box if the answer is true

○   `[] === true`

○   `[[[[[[]]]]]] == ''`

○   `1 < 2 < 3;`

○   `3 > 2 > 1;`

Problem 9. 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.

```
getData(){
   let response = await fetch("http://validApi.edu")
   let parseObject =  response.json()
   return parseObject
}
```


Problem 10. 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;
})
console.log(result)
```


Problem 11. 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 that second or between 10 seconds and 12 seconds.

Problem 12. Consider the following program. What does it print out? If it prints an error write error. This one is tricky so be careful.

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

Problem 13. 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 14. 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.

```
export default function App() {
  constructor(props){
    super(props)
    props = {a: "b"}
}
```

Problem 15. Which of the following are invalid ways to set the state of a JSX element. Shade in all the invalid options. If all options are valid leave them blank.

○
```
this.setState({
  counter: this.state.grade + this.props.increase,
});
```

○
```
this.setState((state, props) => ({
  counter: state.grade + props.increase
}));
```

○
```
let _counter = this.state + this.props.increase
this.setState({counter: _counter});
```

○
```
this.state.grade += this.props.increase
this.setState({counter: this.state.grade});
```

Problem 16. Consider the layout below on the left. Which style object would create the layout on the right. Assume that the assume that elements 2 and 4 are associated with the red style. Shade in all that apply.
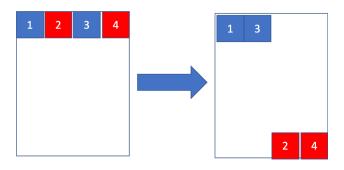

Figure 1: Layout Question 2

○
```
red: {order: 2, alignSelf: 'flex-end'}
```

○
```
red: {order: 2, flexDirection: 'row'}
```

○
```
red: {order: 1, flexDirection: 'column'}
```

○
```
red: {order: -1, alignSelf: 'flex-start'}
```

Problem 17. 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 18. 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 19. Consider the following React Program. What gets logged when the program runs, if nothing gets logged leave the question blank Assume that the program executes without syntax or runtime errors and the component is called by included the following JSX element:

```
<Information type={{a:"alpha"}}/>

export default class Information extends React.Component{
    constructor(props){
     super(props)
     console.log(props)
    }
   render(){
      return (<View/>);
      }
}
```

Problem 20. On my honor I have neither given nor received any aid this exam.