

CS303 FINAL EXAM

October 15, 2020

Name:

The exam takes 2.5 hours.

Please read the exam policy before you start the exam.

Exam Policy:

You can bring one sheet of handwritten notes to the exam. You must turn this sheet and any scratch paper you use in at the end of the exam.

There is a no tolerance policy for dishonesty on exams. **You will be asked to leave the exam room immediately without a warning** for any honesty violations, which mean you will get an **NC**.

Answers should be written with a pen or pencil. If you use a pencil please bring your own eraser and sharpener. You are not allowed to borrow from other students during the exam. If you forgot, ask the instructor or proctor.

All mobile phones should be turned off and stored with your coat or backpack.

You are not allowed to go to the restroom or go out of the room for water after you start the exam.

You are not allowed to ask or get extra papers or anything from other students.

Please write down your answer clearly. If I cannot read your answer, you will not get credit.

Be sure to put your name on every sheet of loose paper that has exam answers. Only hand in paper with exam answers. Do not hand in your sheet of notes or scratch.

Good luck!

[3] What are the 3 actions performed by every constructor function when called with new?

2. [2] Which of these 3 actions are performed if the constructor function is called without new?

3. [2] Suppose `const abc = new class A();` Which of the 3 actions from question 1 are performed?

4. [2] Suppose `const abc = class A();` Which of the 3 actions from question 1 are performed?

5. [2] Suppose `const abc = new class A();` What actions are performed involving `abc.__proto__` in addition to the actions in your answer to question 4?

6. [2] Rewrite the following to use bind instead of call: `const x = foo.call(abc);`

7. [2] Rewrite the following to use bind instead of the wrapper: `const x = function(){ rabbit.run();}`

8. [2] Rewrite the following to use apply instead of call: `const x = foo.call(abc, 1, 2, 3);`

9. [6] On a separate sheet of paper draw the object diagram for all the objects involved in the code below, including the constructor function, prototype, and player objects. You do not need to write the code out for the methods, but put their method names and fn, e.g., aMethod: fn. Be sure to label [[Prototype]] and prototype links.

10. [7] Fill in the blanks below.

11. [4] Rewrite the constructor function on a separate sheet of paper as a class with appropriate methods and properties.

```
function Player(name, age) {
  this.age = age;
  this.name = name;
}

Player.prototype.play = function() {
  return this.name + ' is playing';
}

fred = new Player("Fred", 12);

console.log( fred.age === 12); _____

console.log(fred.__proto__.age === 12); _____

console.log(Player.__proto__.name === "Fred"); _____

console.log(Player.prototype.play() === "Fred is playing"); _____

console.log(fred.play() === "Fred is playing"); _____

pete = new Player("Pete", 15);

console.log(Player.prototype.name === "Pete"); _____

console.log(pete.__proto__ === fred.__proto__); _____
```

12. [6] On a separate sheet of paper draw the object diagram for all the objects involved in the code below, including the constructor functions, prototypes, and surg1 objects. You do not need to write the code out for the methods but put their method names and fn, e.g., aMethod: fn. Be sure to label [[Prototype]] and prototype links.

13. [8] Then fill in the blanks for the console.log lines.

14. [6] Rewrite the code using class syntax with appropriate properties and methods. Do not include any of the console.log statements or any statements such as Person.call that are not needed in class syntax.

```

function Person(name) {
  console.log('Person1 this: ', this); _____
  this.name = name;
  console.log('Person2 this: ', this); _____
}
function Doctor(name, department) {
  console.log( this); _____
  Person.call(this, name, department);
  console.log(this); _____
  this.dept = department;
}
Doctor.prototype.report = function() {
  return "medical report"
};
function Surgeon(name, department) {
  console.log(this); _____
  Doctor.call(this, name, department);
  console.log(this); _____
}
Surgeon.prototype.operate = function() {
  return 'operation performed.'
};
Surgeon.prototype.__proto__ = Doctor.prototype;
surg1 = new Surgeon("Fred", "Cardiology");

console.log(surg1.dept); _____

console.log(surg1.name); _____

console.log(surg1.report()); _____

console.log(surg1); _____

console.log(surg1.__proto__); _____

console.log(surg1.prototype); _____

console.log(Surgeon.__proto__); _____

console.log(Surgeon.prototype); _____
console.log(Surgeon.prototype.__proto__); _____

```