

CS252 end sem exam

27th November, 2019

3 hours

1. I wrote a shell script to do something. Here it is

```
#!/bin/sh

testfun()
{
    echo "Function called with : $@"
    var1=2
}

echo "Script called with $@"
var1=1
testfun 1 2 3
echo "var1 is $var1"
```

As soon as I had written it, I tried to run it and got the following output

```
nisheeth > ./test_shell.sh
bash: ./test_shell.sh: Permission denied
```

(a) What command should I run in CLI to get the script to run? (1 mark)

chmod

(b) What is the most secure permission octet I can use to get the script to run? (2 marks)

500, also give 1 mark if they say 555 or 655

(c) What will be the output of the script, once I do get it to run the way I am trying to? (3 marks)

**Script called with
Function called with : 1 2 3**

var1 is 2

Give +1 for each correct line

2. Here is some C code to do something. Can you write a shell script that does the same thing? (6 marks)

```
#include <stdio.h>

int main()
{
    int i, N, count;

    printf(" Special numbers from 1 to 100 are: \n");
    for(N = 1; N <= 100; Number++)
    {
        count = 0;
        for (i = 2; i <= N/2; i++)
        {
            if(N%i == 0)
            {
                count++;
                break;
            }
        }
        if(count == 0 && N != 1 )
        {
            printf(" %d ", N);
        }
    }
    return 0;
}
```

3. Socket programming is a/an transport layer abstraction because it specifies a communication protocol for host-to-host networking. (2 marks)

4. Consider the following client-server details. (3 marks)

Client host IP address: 103.89.112.3

Server host IP address: 208.216.123.149

You also know that the server is a web server serving HTTP requests. Given this information, specify

- (a) the client socket address 103.89.112.3:some port greater than 1024
- (b) the server socket address 208.216.123.149:80
- (c) the connection socket pair 103.89.112.3:some port greater than 1024, 208.216.123.149:80

5. Fill in the blanks in the following code and state how its meant to be used (7 marks)

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <netinet/in.h>

int main(int argc, char *argv[])
{
    struct addrinfo x, *y, *p;
    int status;
    char ipstr[INET_ADDRSTRLEN];

    memset(&x, 0, sizeof x);
    x.ai_family = _AF_INET_; // AF_UNSPEC also acceptable
    x.ai_socktype = SOCK_STREAM;

    status = getaddrinfo(argv[1], NULL, _&x_, _&y_);

    printf("Results for %s:\n\n", argv[1]);

    for(p = _y_; p != _NULL_; p = p->ai_next) {
        void *addr;
        char *ipver;

        struct sockaddr_in *ipv4 = (struct sockaddr_in *)p->ai_addr;
        addr = &(ipv4->sin_addr);
        ipver = "IPv4";

        _inet_ntop_(p->ai_family, addr, ipstr, sizeof ipstr); // convert to string, worth two points
        printf(" %s: %s\n", ipver, ipstr);
    }

    freeaddrinfo(y);

    return 0;
```

```
}
```

6. I made a simple website as below.

```
<html>
<body>

<form action="test.php" method="post">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="address"><br>
<input type="submit">
</form>

</body>
</html>
```

I want to both display the entered details back to the user, and write these details to a user.log file stored on my server. Can you write the test.php file needed to accomplish both these things? (6 marks)

```
$name = $_POST["name"]
$address = $_POST["address"]
```

```
echo $name
echo $address
```

```
$file = fopen("user.log", "a");
echo fwrite($file, $name . "," . $address);
fclose($file);
```

7. My SQL database, hosted on port 5467 at host IP address 172.10.25.43, holds a database *students*. I started writing a script to pull out students' *names* and *marks* from the table CS252 in this database, but missed some important details. Can you fill in these details so the script works? (10 marks)

```
<!DOCTYPE html>
<html>
<body>
```

```
<?php
```

```
// Create credentials
```

```
$servername = "172.10.25.43:5467";
$username = "username";
```

```
$password = "password";  
$dbname = "students";
```

4 marks

```
$conn = new mysqli($servername, $username, $password, $dbname); 1 mark
```

```
$sql = __SELECT name, marks FROM CS252;__ 2 marks
```

```
$result = $conn->query($sql);
```

```
if ($result->num_rows > 0) {  
    // output data of each row  
    while($row = $result->fetch_assoc()) {  
        __echo " Name: " . $row["name"]. " - Marks:" . $row["marks"]. "<br>";_____  
    } 3 marks  
} else {  
    echo "0 results";  
}
```

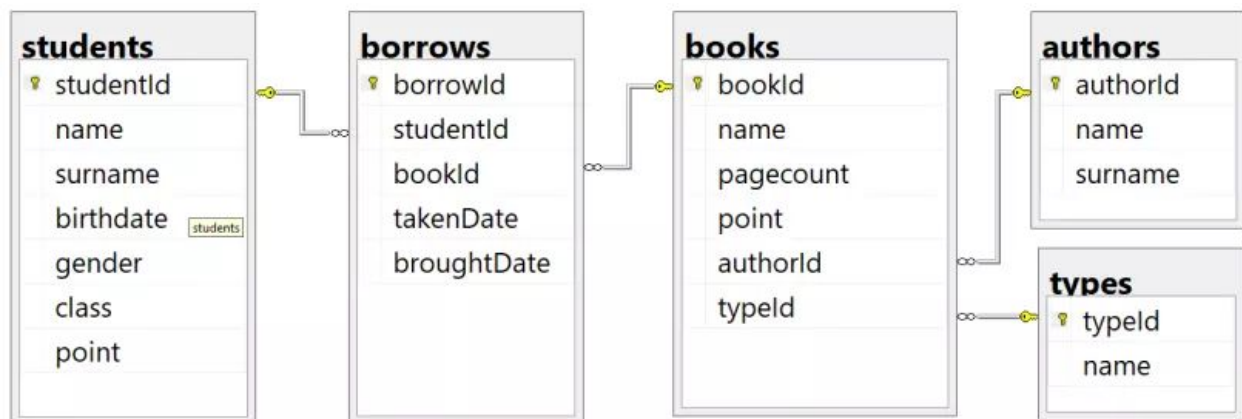
```
$conn->close();
```

```
?>
```

```
</body>
```

```
</html>
```

8. The *library* database has the following relational structure (special token alongside a column denotes it is the primary key for that table)



(a) Can you write an SQL query to list the number of students in each class in the school? (2 marks)

```
Select class,count(*) as StudentCount
from students
group by class
```

(b) Can you write an SQL query to find the name of the youngest child to have borrowed a book from the library? (4 marks)

```
Select name,surname, MIN(birthdate)
from students
join borrows on students.studentId = borrows.studentId
Where borrowId is NOT null
```

(c) Can you write an SQL query to find the most popular author in the library (the one whose books are borrowed the most)? (6 marks)

```
Select name, surname, count(*) as BorrowCount
From authors
Join books on authors.authorId = books.authorId
Join borrows on books.bookId = borrows.bookId
Group by bookId
Order by count descending
```

9. True or false? (4 marks)

- (a) GET requests remain in browser history, but POST requests don't. (T)
- (b) GET requests can be cached, but POST requests can't be (T)
- (c) POST requests encrypt data sent to the server, GET requests transmit in plaintext (F)
- (d) GET requests cannot be infinitely long, but POST requests can be (T)

10. Please convert the XML document below into a JSON object. (3 marks)

```
<items>
  <item id="0001" type="donut">
    <name>Cake</name>
    <ppu>0.55</ppu>
    <batters>
      <batter id="1001">Regular</batter>
      <batter id="1002">Chocolate</batter>
      <batter id="1003">Blueberry</batter>
    </batters>
    <topping id="5001">None</topping>
    <topping id="5002">Glazed</topping>
  </item>
</items>
```

```
        <topping id="5005">Sugar</topping>
    </item>
</items>
```

11. This should look familiar. Its a simplified version of the gibberish API demo from class. I am giving you the HTML file below.

```
<!DOCTYPE html>
<head>

    <title> Gibberish generator </title>
</head>

<body>
    <p> Press to get some gibberish <button id="click" onclick="moreGib()"> here
</button></p>

    <div name="stuff" id="text">
        Text goes here.
    </div>
</body>

<script type="text/javascript" src="script.js"></script>

</html>
```

Now, please look at the JS code associated with the HTML

```
let url = "https://www.randomtext.me/api/gibberish/p-5/10/";
const req = new XMLHttpRequest();
req.open("GET", url);
req.send();
document.getElementById("text").innerHTML = JSON.parse(req.response).text_out;
```

As we discussed in class, this code doesn't work.

(a) Why doesn't it work? (2 marks)

Because the code setting the innerHTML does not wait for the network request to complete

(b) Can you produce code that does work, without using *fetch*? (5 marks)

```

function
get(url){

    return new Promise(function(resolve,
reject){

        var req = new XMLHttpRequest();

        req.open('GET', url);


        req.onload = function(){

            if(req.status == 200){

                resolve(req.response);

            }

            else{

reject(Error(req.statusText));

            }

        };


        req.onerror = function(){

            reject(Error("Network Error"));

        };


        req.send();

    });

}

```

2 points for trying to use some asynchronous code (promise, async/await)

2 points for handling the req variable correctly to receive the response

1 point for handling errors

12.

(a) What will the JS script do (i) when the page loads (assume *par* is a *p* element on the page), and (ii) when the button *btn* is clicked? (4 marks)


```
test = function() {
  document.getElementById("par").innerHTML += this;
}
window.addEventListener("load", test);
document.getElementById("btn").addEventListener("click", test);
```

Print [object Window] on load and [object HtmlButtonElement] on every click in the webpage

(b) Consider the JS code below and answer the following questions (4 marks)

```
function testreturn() {
  return {
    test: 1
  }
}
alert(typeof X);
```

- (i) What will be the output of this code if you replace X by testreturn? **function**
- (ii) What will be the output of this code if you replace X by testreturn()? **object**
- (iii) What will be the output of this code if you replace X by testreturn().test? **number**
- (iv) What will be the output of this code if you don't replace X at all? **undefined**

(c) Consider the JS snippet below and answer the following questions (3 marks)

```
let a = 0.2, b = 0.4, c = 0.6;
const addNum = (x,y) => { return x + y;}
let output = c == addNum(a,b);
console.log(output);
```

- (i) The code shows me that *output* is *false*. Why? **Floating point precision poorly handled in JS**
- (ii) Would the output change if I used *let* instead of *const* to define addNum? **No**
- (iii) What would the output be if I defined *c* as *a+b* in line 1? **true**

(d) Consider the following JS function

```
function returnValues(count) {
  return new Promise(function (resolve) {
    var resultValue = 'Result from call ' + String(count)
    setTimeout(() => resolve(resultValue), Math.random() * 1000)
  })
}
```

```
}
```

(i) What is the difference in the nature of the output you'd expect to see when you run code A, B and C respectively using this function? (2 marks)

```
async function codeA() {  
  for (let i = 1; i <= 10; i++) {  
    var resultValue = await returnValues(i)  
    console.log(resultValue)  
  }  
}
```

```
function codeB() {  
  var promisesArray = []  
  for (let i = 1; i <= 10; i++) {  
    promisesArray.push(returnValues(i))  
  }  
  var allPromises = Promise.all(promisesArray)  
  allPromises.then(resultArray => console.log(resultArray))  
}
```

Code A returns requests to client in the order in which they are received. Code B returns requests all together once they are all fulfilled.

(ii) Can you give examples of web service applications where either of the coding strategies deployed in Code A and Code B would be more useful? (2 marks)

13. Answer the following questions in less than 30 words each (6 marks)

(a) How does a PHP-based web server access data in a text file saved on the server's filesystem?

New process is spawned, master process waits until resource is returned, then resumes

(b) How does a NodeJS based webserver perform the same operation?

Node server generates callback requesting resource and continues with other activities, resource is returned by the callback function to the node process asynchronously

(c) What is the difference between global and local npm installs?

Local npm installs are available within their project folders, global installs are available within the node shell and across that node installation.

14. Consider the following document from the mongodb collection *bios* and answer the following questions.

```
{
  "_id" : 9,
  "name" : {
    "first" : "James",
    "last" : "Gosling"
  },
  "birth" : ISODate("1955-05-19T04:00:00Z"),
  "contribs" : [
    "Java"
  ],
  "awards" : [
    {
      "award" : "The Economist Innovation Award",
      "year" : 2002,
      "by" : "The Economist"
    },
    {
      "award" : "Officer of the Order of Canada",
      "year" : 2007,
      "by" : "Canada"
    }
  ]
}
```

(a) Can you write a Mongo query to find the person who won the Turing Award in 1977, awarded by the ACM? (1 mark)

```
collection.aggregate([ {$match: {awards: {award: "Turing Award", year: 1977, by: "ACM" }}}},
{$project: {_id: 0, name: {last:1}}}]
```

(b) Can you list all the awards won by Grace Hopper? (2 marks)

```
collection.aggregate([ {$match: {name: {first: "Grace", last: "Hopper" }}}}, {$project: {_id: 0,
awards: {award:1}}}]
```

(c) Can you write a Mongo query to find the youngest person in this collection? (3 marks)

```
collection.aggregate([ {$sort: {birth: -1}}, {$project: {_id: 0, name: {last:1}}}, {$limit: 1}])
```

15. Look at the React code below and answer the following questions

```

class TestClass extends React.Component {
  constructor(props) {
    super(props);
    this.state = {grade: "B"};
  }
  componentDidMount() {
    setTimeout(() => {
      this.setState({grade: "A"})
    }, 1000)
  }
  render() {
    return (
      <h1>My CS252 grade is {this.state.grade}</h1>
    );
  }
}

```

ReactDOM.render(<TestClass />, document.getElementById('root'));

(a) What will be the behavior of the webpage that this script renders? (2 marks)

It will show "My CS252 grade is B" and replace with "My CS252 grade is A" after 1 second

(b) Can you edit this code to use the props variable rather than the state variable, such that the website behaves the same way as before? (2 marks)

No

(c) If I want **TestClass** to affect only one part of my webpage, without affecting the operation of other parts of the page, how would I change the code above? (2 marks)

Replace root with some sub-element

(d) Which of these are front-end frameworks: Vue, Meteor, Ruby on Rails? (1 mark)

Vue