

Week 3

Exercise 1:

Output:

```
*** THIS IS MY FIRST PROGRAM ***  
Welcome Anakin, Your monthly salary is £80000.
```

Explanation:

I defined a function with two parameters that printed a message and then called it.

Exercise 2:

Output:

```
Expert in Java.
```

Explanation:

I created an arrow function with one parameter that printed a message and then called it.

Exercise 3:

Output:

```
Student name: Antony  
Location: UEL Campus  
Date of birth: 12/12/1980  
Grade is A grade  
Grade is B grade  
Using Person module: [ 'Jim', 'USA', 'myemail@gmail.com' ]  
*** PROGRAMME END ***
```

Explanation:

I created two additional JS files, exported the various functions and class within, and then imported them into the index file. I called the various functions and printed the results to the console.

Exercise 4:

Output:

```
Temporary directory: C:\Users\Anton\AppData\Local\Temp
Hostname: Skynet-laptop
OS: win32, release: 10.0.22631
Uptime: 87.98581138888888 hours
User info: {
  uid: -1,
  gid: -1,
  username: 'Anton',
  homedir: 'C:\\Users\\Anton',
  shell: null
}
Memory: 16.872415232 Gigabytes
Free: 8.074645504 Gigabytes
CPU: [
  {
    model: '11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz',
    speed: 2995,
    times: {
      user: 1783640,
      nice: 0,
      sys: 1739828,
      idle: 58860937,
      irq: 343109
    }
  },

```

```
  {
    model: '11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz',
    speed: 2995,
    times: { user: 1177859, nice: 0, sys: 728486, idle: 60477250, irq: 42125 }
  },
  {
    model: '11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz',
    speed: 2995,
    times: { user: 1433437, nice: 0, sys: 595468, idle: 60354625, irq: 15203 }
  },
  {
    model: '11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz',
    speed: 2995,
    times: { user: 888859, nice: 0, sys: 376031, idle: 6118625, irq: 6750 }
  },
  {
    model: '11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz',
    speed: 2995,
    times: { user: 612656, nice: 0, sys: 340750, idle: 61430093, irq: 9187 }
  },
  {
    model: '11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz',
    speed: 2995,
    times: { user: 479296, nice: 0, sys: 267437, idle: 61636781, irq: 6437 }
  },
  {
    model: '11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz',
    speed: 2995,
    times: { user: 577531, nice: 0, sys: 301075, idle: 61504078, irq: 7312 }
  },
  {
    model: '11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00GHz',
    speed: 2995,
    times: { user: 486375, nice: 0, sys: 231343, idle: 61665750, irq: 3781 }
  }
]
```

```
Network: {
  'Wi-Fi': [
    {
      address: [REDACTED],
      netmask: 'ffff:ffff:ffff:ffff::',
      family: 'IPv6',
      mac: [REDACTED],
      internal: false,
      cidr: [REDACTED],
      scopeid: 10
    },
    {
      address: [REDACTED],
      netmask: '255.255.255.0',
      family: 'IPv4',
      mac: [REDACTED],
      internal: false,
      cidr: [REDACTED]
    }
  ],
  'Loopback Pseudo-Interface 1': [
    {
      address: '::1',
      netmask: 'ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff',
      family: 'IPv6',
      mac: '00:00:00:00:00:00',
      internal: true,
      cidr: '::1/128',
      scopeid: 0
    },
    {
      address: '127.0.0.1',
      netmask: '255.0.0.0',
      family: 'IPv4',
      mac: '00:00:00:00:00:00',
      internal: true,
      cidr: '127.0.0.1/8'
    }
  ]
},
```

```

'vEthernet (Default Switch)': [
  {
    address: [REDACTED],
    netmask: 'ffff:ffff:ffff:ffff:',
    family: 'IPv6',
    mac: [REDACTED],
    internal: false,
    cidr: [REDACTED],
    scopeid: 24
  },
  {
    address: [REDACTED],
    netmask: '255.255.240.0',
    family: 'IPv4',
    mac: [REDACTED],
    internal: false,
    cidr: [REDACTED]
  }
]
}
*** PROGRAMME END ***

```

Explanation:

For this exercise, I used the core modules `os` and `util` to gather and print system information to the console.

Reflection:

This week I learned how to construct and call both regular and arrow functions in JavaScript. I also learned about modules: how to create local ones and access core ones. In particular, the keyword `export` is used to define which code is to be accessible outside of the native file. I acquired knowledge on the various methods of the core `os` module which I then put into practice to print various system information about my computer. Overall, I believe this week has helped me to acquire a better understanding of how JavaScript code is organised, managed, and distributed across many applications.