# Develop a complex computer program for a specified task

## Variables – scope – data type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable Name** | **Data Type** Integer, String | **Scope** | **Example** | **Description** |
| data | Object | Global | var data = {  stops: {…}  timesMonFri : {…}  timesFri: {…}  timesSat: {…}  timesSun: {…}  } | Holds all of the data for the bus times that they depart as well as the stops |
| buses | Object | Global | var buses = ["awapuni",  "rugby",  "highbury",  "takaro",  "cloverlea",  "milson",  "rhodes",  "roslyn",  "rangiora",  "brightwater",  "fernlea",  "heights",]; | Creates the array of the bus names that can be used later for being clicked on. |
| date | Interger | Local to getDay() function | var date = new Date(); | Stores the date into a variable |
| day | Interger | Local to getDay() function | var day = date.getDay(); | Stores the day in a variable |
| friO | Object | Local to getDay() function | var friO = [];  friO = this.friTimes | Gets the array of friTimes and stores into a new variable |
| fri | Object | Local to getDay() function | var fri = [];  fri = this.monFriTimes | Gets the array of Monday Friday times and stores in a variable |
| html | String | Local to getStops() function | var html = '<ul>';  for (var i = 0; i < this.stops.length; i++) {  html += "<li class='busStops' onclick=\"routes['" +this.name.toLowerCase() + "'].showStops(" + i + ")\">" + this.stops[i] + "</li>";  }  html += '</ul>';  return html; | Stores html code of bus stops to be put into the DOM |
| i | Interger | Local to getStops() function  Local to getTimes() function  Local to showNext() function  Local to createMarker() function | for (var i = 0; i < this.stops.length; i++) {  for (var i = 0; i < day.length; i++) {  for (var i = 0; i < day.length; i++) {  for (var i = 0; i < this.buses.length; i++) { | Loops over using a set variable until the right piece is found |
| times | String | Local to getTimes() function | var times = '';  times += "<li class='tinmesRow'>" + day[i][index].toFixed(2) + "</li>"; | Empty string that then stores the timeof the buses |
| h | Interger | Local to showNext() function | var h = date.getHours() | Stores the time in hours as a new variable |
| m | Interger | Local to showNext() function | var m = date.getMinutes() | Stores the time in minutes as a new variable |
| time | Integer | Local to showNext() function | var time = h + "." + m | Joins the hours and mintures times together |
| next | String / Interger? | Local to showNext() function | var next = day[i][index].toFixed(2)  var next = "Sorry no more busses for today"  return next | Stores the result after a for loop for when the next bus is. |
| stopPosition | Object | Local to createMarker() function | var stopPosition = this.stopPositions[index]; | Stores as the bus that is clicked on to get the position of what bus is clicked on |
| stopName | Object | Local to createMarker() function | var stopName = this.stops[index]; | Stores as the bus is clicked on to get the stops to show for the particular bus |
| routes | Object | Local to createMarker() function | var routes = {};  routes[name] = newRoute; | Stores the routes as the object as the. Bus that is clicked on and all the information that goes with that bus |
| name | String | Local to createMarker() function | var name = buses[i]; | Stores the bus name that is clicked on. To be recalled later |
| newRoute | Object | Local to createMarker() function | var newRoute = new Bus(  name,  data.stops[name], data.stopCoordinates[name],data.timesMonFri[name],  data.timesFri[name],  data.timesSat[name],  data.timesSun[name]  ); | Stores the information for the bus that has been clicked on |
| theSquare | Integer | Global | var theSquare = {  lat: -40.356207,  lng: 175.610062  }; | Stores the latitude and longitude for the square to be used as the centre of the map on page load |

## Index Data Structure

|  |  |  |  |
| --- | --- | --- | --- |
| **Array Name** | **Scope** | **Example** | **Description** |
| buses | Global | var buses = [ "awapuni", "rugby", "highbury", "takaro", "cloverlea", "milson", "rhodes", "roslyn", "rangiora", "brightwater","fernlea", "heights"]; | This array stores all the names of the buses. It is used for lots of things such as the amount of times we have buses |
| data.stops.awapuni | Global | awapuni: [  "Depart MST",  "Park Rd",  "College Street",  "Pitman Rd",  "Maxwells Line",  "College Street",  "Rugby Street",  "Ferguson Street",  "Arrive MST"  ], | This Array stores all the names of the streets that the bus stops at. It is used to show where they stop on the map when been clicked on. |
| **^^This is the same for all the other stops. But all with different stop names^^** | | | |
| Data.stopCoordinates.awapuni | Global | awapuni: [  new google.maps.LatLng(-40.35544, 175.61281),  new google.maps.LatLng(-40.36957, 175.608092),  new google.maps.LatLng(-40.370013, 175.598573),  new google.maps.LatLng(-40.374685, 175.594374),  new google.maps.LatLng(-40.379603, 175.584714),  new google.maps.LatLng(-40.375575, 175.58587),  new google.maps.LatLng(-40.370879, 175.586738),  new google.maps.LatLng(-40.365486, 175.5999),  new google.maps.LatLng(-40.35544, 175.61281)  ], | This Array stores all latitude and longitude where the bus stops are located. It is usewd to show on the map where these stops are. |
| **^^This is the same for all the other stops. But all with different latitude and longitudes^^** | | | |
| data.timesMonFri.awapuni | Global | awapuni: [  [7.00, 7.08, 7.11, 7.13, 7.15, 7.19, 7.22, 7.25, 7.35],  [7.40, 7.48, 7.51, 7.53, 7.55, 7.59, 8.02, 8.05, 8.15],  [8.20, 8.28, 8.31, 8.33, 8.35, 8.39, 8.42, 8.45, 8.55],  [9.00, 9.08, 9.11, 9.13, 9.15, 9.19, 9.22, 9.25, 9.35],  [10.20, 10.28, 10.31, 10.33, 10.35, 10.39, 10.42, 10.45, 10.55],  [11.40, 11.48, 11.51, 11.53, 11.55, 11.59, 12.02, 12.05, 12.15],  [13.00, 13.08, 13.11, 13.13, 13.15, 13.19, 13.22, 13.25, 13.35],  [14.20, 14.28, 14.31, 14.33, 14.35, 14.39, 14.42, 14.45, 14.55],  [15.20, 15.28, 15.31, 15.33, 15.35, 15.39, 15.42, 15.45, 15.55],  [16.00, 16.08, 16.11, 16.13, 16.15, 16.19, 16.22, 16.25, 16.35],  [16.40, 16.48, 16.51, 16.53, 16.55, 16.59, 17.02, 17.05, 17.15],  [17.20, 17.28, 17.31, 17.33, 17.35, 17.39, 17.42, 17.45, 17.55],  [18.00, 18.08, 18.11, 18.13, 18.15, 18.19, 18.22, 18.25, 18.35],  [24.00, 23.02, 23.03, 23.04, 23.05, 23.06, 23.07, 23.08, 23.09]  ], | This array stores all of the times that the bus arrives at the stops between Monday and Friday. It is used to display the time for the stop that has been clicked on by the user. |
| **^^This is the same for all the other stops. But all with different stop names^^** | | | |
| data.timesFri.rugby | Global | rugby: [  [18.40, 18.46, 8.49, 18.52, 18.55, 18.58, 19.02, 19.05, 19.15],  [20.20]  ], | This Array stores the times for the buses that only run on a Friday. It is used to show the time of when the bus will arrive at the stop that has been clicked on. |
| **^^This is the same for all the other stops. But all with different stop names^^** | | | |
| data.timesSat.awapuni | Global | awapuni: [  [9.00, 9.08, 9.11, 9.13, 9.15, 9.19, 9.22, 9.25, 9.35],  [11.40, 11.48, 11.51, 11.53, 11.55, 11.59, 12.02, 12.05, 12.15],  [14.20, 14.28, 14.31, 14.33, 14.35, 14.39, 14.42, 14.45, 14.55],  [15.00, 15.08, 15.11, 15.13, 15.15, 15.19, 15.22, 15.25, 15.35],  [16.20]  ], | This array stores the times for the buses that run on Saturday. It is used to show the time of when the bus will arrive at the stop that has been clicked on |
| **^^This is the same for all the other stops. But all with different stop names^^** | | | |
| data.timesSun.awapuni | Global | awapuni: [  [8.20, 8.28, 8.31, 8.33, 8.35, 8.39, 8.42, 8.45, 8.55],  [12.20, 12.28, 12.31, 12.33, 12.35, 12.39, 12.42, 12.45, 12.55],  [16.20]  ], | This array stores the times that the buses will run on Sunday. It is used to display the times when the stop is clicked on |
| **^^This is the same for all the other stops. But all with different stop names^^** | | | |
| friO | Local to the getDay() function | var friO = []; | This array is empty but gets filled with the times for the Friday only bus. It is used to show the times that the bus will arrive on a Friday |
| fri | Local to the getDay() function | var fri = []; | This array is empty bus gets filled with the Monday to Friday bus times as these are different to the busses that fun on a Friday only. It is used to display the times that the busses arrive. |

## Modular Structure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | **Scope** | **Procedural Structure (Steps)** | **Example** | **Description** |
| **All of the fucntions are in the constructor so they are listed below** | | | | |

## Class / Objects

|  |  |
| --- | --- |
| **Class** | |
| **Data:** (variables) | |
| Name: | Description: |
| name | This is the name of the bus |
| stops | This is the stops that the buses will stop at |
| stopPositions | This is the coordinates that the buses will stop at. They will be shown on the map. |
| monFriTimes | This is the times that the buses will stop at the selected stop |
| friTimes | This is the times for the Friday only busses that will stop at the stop |
| satTimes | This is the times for the Saturday buses that will stop at the selected stop |
| sunTimes | This is the times for the Sunday buses that will stop at the selected stop |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Behaviour:** (functions) | | | | |
| **Function Name** | **Scope** | **Procedural structure (Steps)** | **Example** | **Description** |
| getDay | Global | 1. Create function to get the day 2. Get the date 3. If the day is 1,2,3or4 get times for Monday Friday 4. If day is 5 then get times for Monday Friday and Friday. 5. If day id 6 then get the times for Saturday only 6. If it does not come back with a day then display the times for Sunday 7. Return the day number which will show the times for the specific stop | getDay() {  var date = new Date();  var day = date.getDay();  if (day == 1 || day == 2 || day == 3 || day == 4) {  day = this.monFriTimes;  } else if (day == 5) {  var friO = [];  friO = this.friTimes  var fri = [];  fri = this.monFriTimes  var day = fri.concat(friO);  }else if (day == 6) {  var day = this.satTimes;  }else {  var day = this.sunTimes;  }  return day;  } | This function gets the date checks it to see what number day it is. Then respond the times for the day. |
| getStops | Global | 1. Create a function to get the stops 2. Create variable to begin the unordered list tag 3. Loop over the length of the stops. 4. Add each stops array to the HTML string as an html list item with an on click even to call the function to show the stops. 5. End the unordered list tag. 6. Return the variable of the unordered list which will now show all the information. | getStops() {  var html = '<ul>';  for (var i = 0; i < this.stops.length; i++) {  html += "<li class='busStops' onclick=\"routes['" +this.name.toLowerCase() + "'].showStops(" + i + ")\">" + this.stops[i] + "</li>";  }  html += '</ul>';  return html;  } | This loops over the length of the stops and gets the route name to show the stops that are available. |
| getTimes | Global | 1. Create function to get the tinmes from the index 2. Create a empty variable 3. Create a variable that will get the day 4. Loop over the day to check each stop time and if there is nothing in the array place it will not show anything. 5. If there is a number then display the number | getTimes(index) {  var times = '';  var day = this.getDay()  for (var i = 0; i < day.length; i++) {  if (day[i][index] == null || day[i][index] == undefined){  times += ""  }  else {  times += "<li class='tinmesRow'>" + day[i][index].toFixed(2) + "</li>";  }  }  return times;  } | Gets the times of the stop arrays and if there is nothing in the stop time that should show for the home only buses then it will snot show anything. |
| showNext | Global | 1. Create a function to show the next bus from the index 2. Create a variable to ge the day 3. Create a variable to get the date 4. Create a variable to get the house 5. Create a cariable to get the monutes 6. Carea a variable that joins the hours and minutes together to make a time. 7. Loop over the length of the dayand if the day and if the times is before the next time then display the time. 8. If the time is not before the last bus then display that there are no more buses for the day | showNext(index) {  var day = this.getDay()  var date = new Date();  var h = date.getHours()  var m = date.getMinutes()  var time = h + "." + m  for (var i = 0; i < day.length; i++) {  if (time < day[i][index]) {  var next = day[i][index].toFixed(2)  return next  }  else {  var next = "Sorry no more busses for today"  return next  }  }  this.nextDOM = next  } | This function gets the day and time in hours and minutes to make the current time. Then check to see what day it is and then if the time is before the next bus then show the next closest one. If there is no bus then it will show that there are no more buses for the day. |
| showStops | Global | 1. Create a function to display the stops 2. Set the route name from the HTML to the name of the bus 3. Set the stop from the HTML to the stops from the index of the route that has been clicked on 4. Set the createMarker function to the stop that has been chosen. 5. Set the times of the bus that has been clicked ion to display when the bus is clicked on in the getTimes function. 6. Set the times of the next bus to display when the bus has been clicked on. | showStops(index) {  this.routeNameDOM.innerHTML = this.name;  this.stopsDOM.innerHTML = this.stops[index];  this.createMarker(index);  this.timesDOM.innerHTML = this.getTimes(index);  this.nextDOM.innerHTML = this.showNext(index)  } | This function puts all of the information from the functions onto the correct place on the page. |
| createMarker | Global | 1. Create a function to create the marker 2. If there a marker previously on the map then set the map and marker to have nothing displayed. 3. Create variable to get the position of the stop that has been clicked on. 4. Create a variable to get the stop name 5. Set the marker to be a new marker on the map at the stop position that has been clicked on 6. Set the zoom so that user can see the stop clearly. | createMarker(index) {  if (this.marker != null) {  this.marker.setMap(null);  this.marker = null;  }  var stopPosition = this.stopPositions[index];  var stopName = this.stops[index];  this.marker = new google.maps.Marker({  map: map,  position: stopPosition,  title: stopName  });  map.setCenter (stopPosition);  map.setZoom(15);  } | This function creates the marker that is displayed on the map when the user clicks on a bus stop. Making sure also that there is only one marker on the map at a time |

|  |  |
| --- | --- |
| **New Bus objects** | |
| **Name:** | **Example:** |
| routes.awapuni | {name: "awapuni",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(14),  friTimes: Array(1)…} |
| routes.rugby | {name: "rugby",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(13),  friTimes: Array(2)…} |
| routes.highbury | {name: "highbury",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(13),  friTimes: Array(0)…} |
| routes.takaro | {name: "takaro",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(13),  friTimes: Array(1)…} |
| routes.cloverlea | {name: "cloverlea",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(20),  friTimes: Array(2)…} |
| routes.milson | {name: "milson",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(21),  friTimes: Array(1)…} |
| routes.rhodes | {name: "rhodes",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(13),  friTimes: Array(2)…} |
| routes.roslyn | {name: "roslyn",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(13),  friTimes: Array(1)…} |
| routes.rangiora | {name: "rangiora",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(13),  friTimes: Array(2)…} |
| routes.brightwater | {name: "brightwater",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(13),  friTimes: Array(1)…} |
| routes.fernlea | {name: "fernlea",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(13),  friTimes: Array(1)…} |
| routes.heights | {name: "heights",  stops: Array(9),  stopPositions: Array(9),  monFriTimes: Array(13),  friTimes: Array(2)…} |

## Testing and Debugging

*Note: Expected cases are highlighted in Green*

*Note: Boundary cases are highlighted in Yellow*

*Note: Invalid cases are highlighted in Red*

|  |  |  |  |
| --- | --- | --- | --- |
| **What I’m testing** | **Input** | **Expected** | **Result** |
| Testing when the route is clicked on | Clicked on the Awapuni route | All other stops to slide down and reveal the stops that the bus will stop at. | I did not get the result I was expecting so I edited the code and was testing each change until I got it to work as expected |
| Tested when the bus stop was clicked on | Clicked on the route then clicked on the first bus stop | When scrolling down the information would be filled out as it should be with correct times | I did not get the result I wanted as only some of the information showed up. I edited the code so most of the other information showed up |
| Tested the map marker | Clicked on the route then the first bus stop, then checked the functionality of the map | When scrolled down to the map there should be a marker showing the bus stop on the map. | I did not get the result that I wanted as the marker did not show up on the map however the functionality of the map was good.  I then changed some code and got the marker to show up in the correct spot. |
| Tested the previous marker disappearing when clicking on a new stop | Clicked on the route then the second bus stop, then searched the map for only having one marker | When scrolled down to the map there would only be one marker on the map showing when zooming out | This worked as when I zoomed out on the map I was only able to find one marker |
| Tested that the next bus was the correct time to the next bus to arrive at the stop | Clicked on the route then the first bus stop | When scrolled down to the information I checked that the time that was showing under the “Next bus” title matched the time according to the timetable book. | This worked as the time that was displayed matched that time according to the timetable book. |
| Tested what happened when the real time was after the time of the last bus. | Set time to a time later than the last bus stopping at the Awapuni first stop. | I expected the tie to show undefined. | This did happen however I changed this as it was not very helpful to the user as they could have though that it was just a fault.  I then changed it so that instead of saying “undefined” it would tell the user that there were no more buses for the day. |
| Tested that you were unable to have more than one marker showing on the map at the same time | Clicked on one stop and then checked the marker, then clicked on the next stop, zoomed out | I expected that the previous marker that was clicked on would disappear so that there was only one marker on the map making it easier for users | This is what happened, when I zoomed out a little there was only one marker to be seen. |
| Due to this being a click only app (No other user input like typing) there are not many boundary cases that can be tested. | | | |
| Tested to see if there was a possibility of inputting invalid cases and because this is a click only app (No other user input like typing) there is no way of having invalid cases | | | |

## Program Plan *(Pseudocode or Flowchart goes here)*

Start of the program

Create a variable array for all the buses to go into.

Make a bus class that will hold the information of the Name, Stops, Stop Positions, Monday to Friday times, Friday only times, Saturday times, Sunday times.

Gets the elements for document objects and links them to the HTML

Create a function that will get the day using variables.

If statements that will get the day and compare it to check what the number in the array is, if the day is between 1 and 4 then it will return the day variable to be the array for Monday to Friday times.

If the day is returned as 5 then it will combine the Friday only times with the Monday to Friday times to be one array that will become the day variable

If the day is returned as 6 then it will get the array for Saturday times and then become the day variable

If the day is returned as anything else, (the only option for this is 0) it will then get the array for Sunday times and that will then become the day variable.

Just before the end of the function the if statement will return the “Day” that is is as an array to be used later.

Create a function that will get the stops that the busses stop at on the routes.

Create a variable that will run some HTML for an onClick.

Create a loops that will loop over the amount of routes that are there and select the one that has been clicked on to then show the stops that are available.

Return the output of the selected route stops.

Create a function to get the times of the bus stop that has been clicked on.

Create variable called “times” that is empty.

Create variable called “day” that will get the day from the function “getDay”.

Loop over the length of the day.

If the length of the day is null or undefined then set the times variable to display nothing.

If it is not displaying nothing then it will display the times for the busses in a row making sure that there are two number after the bullet point.

The function will then return the outcome of the conditions abouve

Create a function to show the time of the next bus that will arrive.

Create a variable to het the day from the “getDay” function.

Create a variable to get the date

Create a variable to get the hour of the day

Create a variable to get the minutes of the day.

Create a variable to join together the hours and minutes to show the time correctly.

Loop over the length of the day to see what day number of the week it is.

If the time of the day is less than the next bus then display the time of the next bus making sure that there are two number after the decimal point.

If there are no more buses for the day then display that there are no more buses for the day.

Change the “next” element to show the right time

Create a function to show the information on the page.

Change the route name to be the route that is clicked on.

Change the stops name to show the stop that has been clicked on

Run the function to create the marker

Change the times space to now show the times that have been sorted from the function to get the times.

Change the next bus space to show the result of the function that figured out what time bus was next.

Create a function that will make the marker show on the map.

If there is already a marker on the map then get rid of it and centre the map back to the square.

Create a variable to get the stop position of the bus stop that has been clicked on.

Create a variable to get the name of the stop that has been clicked on.

Set the marker to be displayed on the map at the stop position of the stop name.

Set the marker to be in the middle of the page

Set the zoom so that the user can see what is around the bus stop.

Create an empty variable to put all of the route information into.

Loop over the amount of routes that there are to find the one that has been clicked on.

Create a variable called “name” that is the bus that has been clicked on.

Create a variable to construct the bus information. Inside will be:

Name of the bus that has been clicked on

The stops that correspond to the bus that has been clicked on

The Monday through to Friday times of the bus that has been clicked on.

The Friday only times of the bus that has been clicked on.

The Saturday times of the bus that has been clicked on

The Sunday times of the bus that has been clicked on.

Set the empty variable to hold the information of the bus that has been clicked on.

When document is loaded

Create a variable for the location of the square with the latitude and longitude inside.

Sets the map to reload at the centre which is the square with a zoom.

Hide the stops menu

When buses are clicked on

Run the function to get stops.

When clicked on slide down to show the information

End of the program

## Incremental evelopement *(how did your program develop over time. Show version control commits)*