Computing Science (Advanced Higher): project

Candidate evidence

Candidate 1 evidence

Advanced Higher Computing Science Project

Analysis

Description of the problem

I plan to build a dynamic website for building PCs that interacts with a database that holds information on parts such as CPU, motherboards and RAM. The Major focus is the Website that uses html for content, CSS for style and php for interacting with a sql database and minor focus is the Database.

The database will store the part cost, size, brand, information, power and compatibility with parts. PHP code will be able to query this database for power requirements, add up the power demand of the parts and compare it to the power supply. It would also have PHP code to calculate and display the total price of the build. After the build is finished a user can name the build and can insert it into a table that can then be viewed with other user submitted builds.

Scopes and boundaries

Scope: This development involves

- a website that interacts with a database
- Analysis consisting of research, identifying subtasks,
- Wireframes of each webpage and form with clear indication of what everything is.
- a detailed navigational design of the website structure for all pages in the site
- A design of all tables used in the database and a data dictionary.
- a working website and database
- the results of testing
- an evaluation reports
- Names of builds will be validated so that each new build must have a unique name

Boundaries

- Media queries will be used for different screen sizes, but I will not be able to test the site on all OSes, devices and browsers.
- There is no sign in process so anyone accessing the site can view builds and add their own
- The final site will not be live on the internet, it will only be available on localhost on my own laptop
- Prices and details of parts in the database will not be updated dynamically, I will create and populate the database and use those values throughout the project
- It is not a real online shop so users cannot order their components

Constraints

- All work must be completed by 23rd March. Any elements not completed will not be implemented and I will cover this in my evaluation
- I will use Atom for coding the site which is an open-source editor
- I will use EasyPHP to create a local server for testing my site and connecting the page to the database

End-user and Functional requirements

1. (a) A user must be able to choose, from the home page, to read information, to build a new system or to view existing builds
   (b) A functional requirement is that the home page has a navigation bar with links to the main pages

2. A user must be able to read the information page

3. (a) A user must be able to choose a motherboard for their build
   (b) A functional requirement is a drop-down menu with the size and socket types available

4. (a) A user must be able to choose a CPU
   (b) A functional requirement is a drop-down menu with CPU brands

5. (a) A user must be able to choose a case
   (b) A functional requirement is a drop-down menu with Case sizes

6. (a) A user must be able to choose a graphics card
   (b) A functional requirement is a drop-down menu with VRAM and brand

7. (a) A user must be able to submit their choices
   (b) A functional requirement is a submit button that sends data securely to a page that calculates and displays total power requirements and cost

8. (a) If a user doesn't make any choices and just submits their build, they will be assigned the default choices and still given their build
   (b) A functional requirement is that each drop-down menu has a default value that is passed to the final build page even if the user doesn't change any values

9. (a) An end-user requirement would be that users are able to search for parts of a certain brand or type.
   (b) A functional requirement would be that the website would query the database in order to get the information for parts

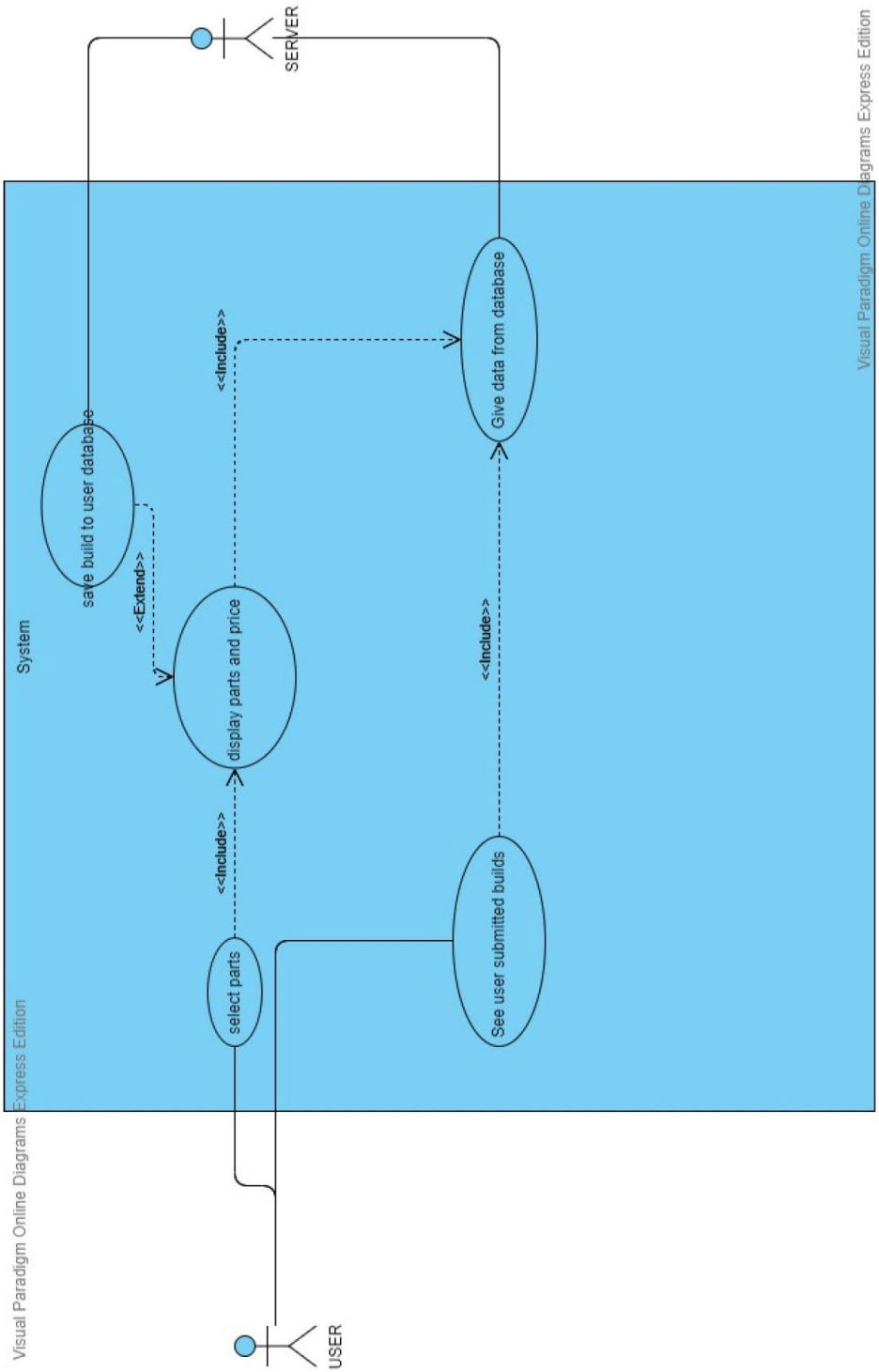
10. (a) A functional requirement would that the user can submit builds created into the user table and see them on another page.
    (b) A functional requirement would be that duplicate names cannot be submitted into the database.
    (c) A functional requirement is that depending on whether a name already exists a user will be sent to either a page asking to re-enter a name, or the name and information will be submitted into the database.
    d) A functional requirement is that a user cannot submit only special characters or numbers as Build name

11. (a) An end user requirement is that the user can see an appropriate view of the webpage regardless of the device, screen size or browser used to access it.
    (b) A functional requirement would be that the web page can fit many different screen and device sizes using media queries.

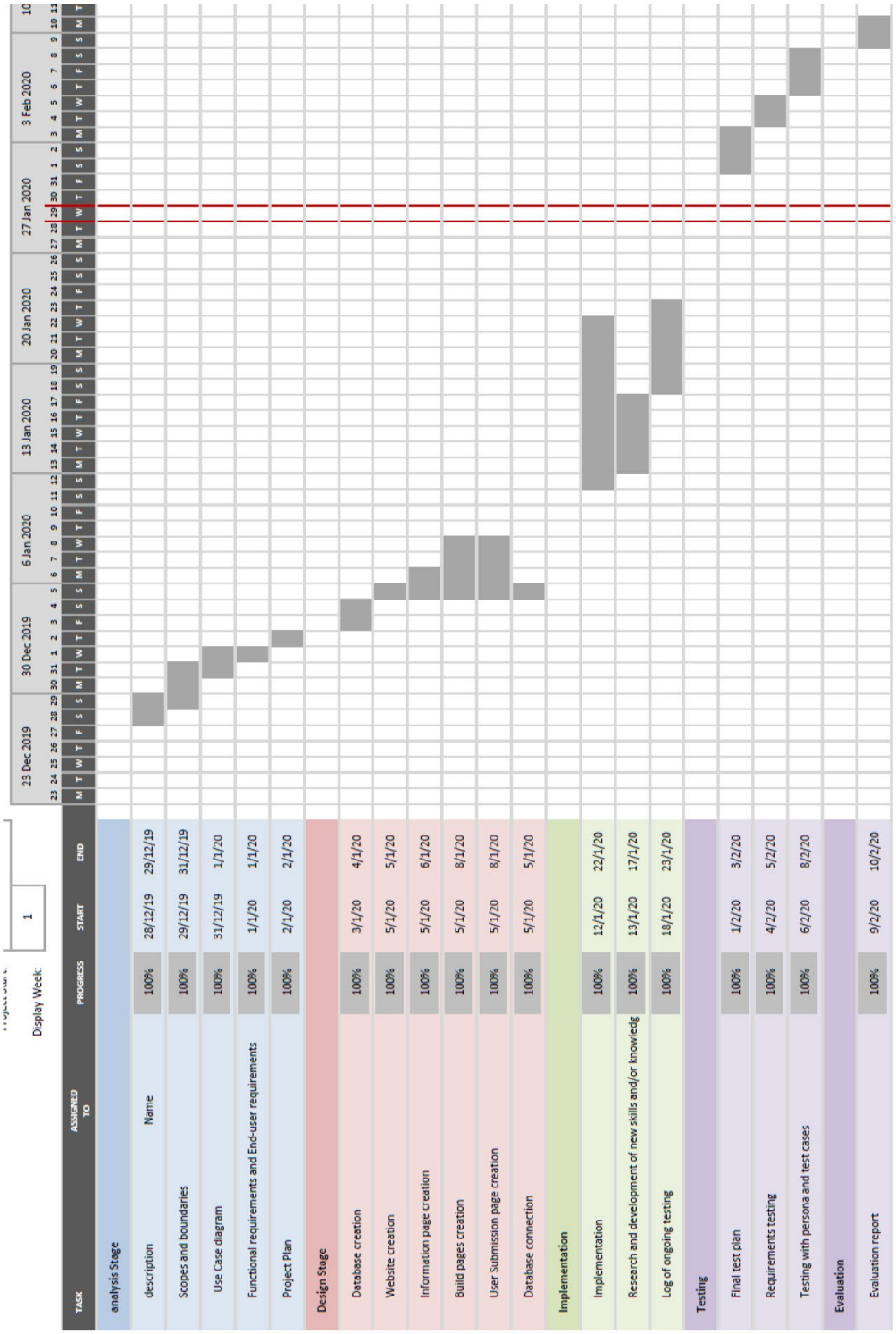
12. A user requirement is that a user can select a Build name and then see all information of parts in said build.

13. (a) A user requirement is that a user can print a hard copy of their choices
    (b) A functional requirement is that the site will allow you to print your selected build showing all the component details but hiding the navigation bar using a print media query

Use case diagram



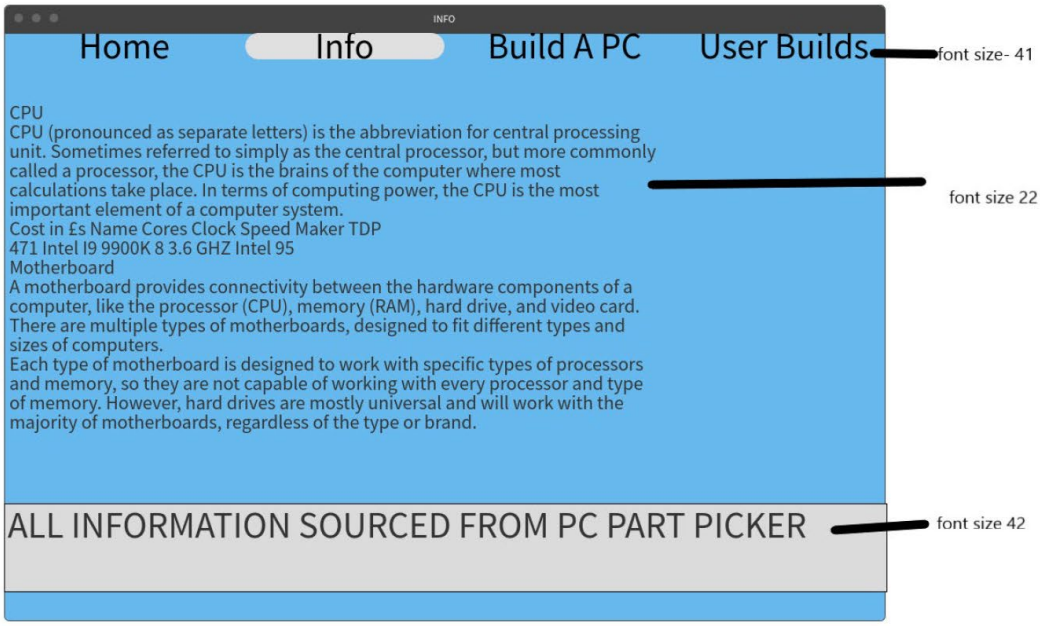
Project Plan



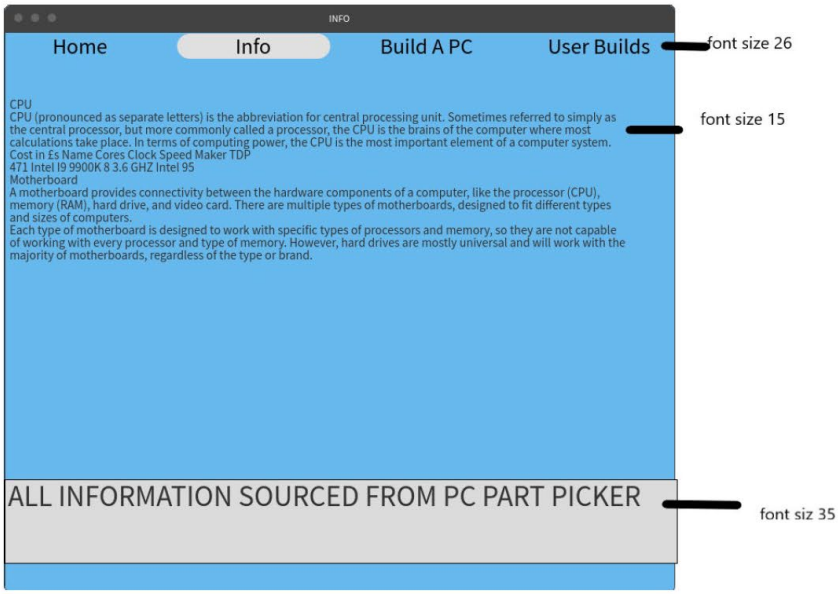
TASK	ASSIGNED TO	PROGRESS	START	END
analysis Stage				
description	Name	100%	28/12/19	29/12/19
Scopes and boundaries		100%	29/12/19	31/12/19
Use Case diagram		100%	31/12/19	1/1/20
Functional requirements and End-user requirements		100%	1/1/20	1/1/20
Project Plan		100%	2/1/20	2/1/20
Design Stage				
Database creation		100%	3/1/20	4/1/20
Website creation		100%	5/1/20	5/1/20
Information page creation		100%	5/1/20	6/1/20
Build pages creation		100%	5/1/20	8/1/20
User Submission page creation		100%	5/1/20	8/1/20
Database connection		100%	5/1/20	5/1/20
Implementation				
Implementation		100%	12/1/20	22/1/20
Research and development of new skills and/or knowledg		100%	13/1/20	17/1/20
Log of ongoing testing		100%	18/1/20	23/1/20
Testing				
Final test plan		100%	1/2/20	3/2/20
Requirements testing		100%	4/2/20	5/2/20
Testing with persona and test cases		100%	6/2/20	8/2/20
Evaluation				
Evaluation report		100%	9/2/20	10/2/20

Design

Normal webpage wireframe



800px wireframe



PHP pseudocode

Connect to database

$username = "root";

$password = "";

$servername = "localhost";

$dbname = "Parts";

$conn = connect($servername, $username, $password, $dbname);

Build a PC PHP

Get CPU, Motherboard, GPU and Case from form

Display all where CPU from CPU table = CPU

Display all where Motherboard from Motherboard table = Motherboard

Display all where GPU from GPU table = GPU

Display all where CASE from CASE table = CASE

Select random RAM row From RAM Table where RAM amount <= Max Ram from Motherboard and display all

Calculate total power demand from CPU, GPU

Itemprice = partprice

Total price = partprice1 + partprice2 + partprice3 + partprice4 +partprice 5 + partPrice6

Display total price

Insert into user build table

Data dictionary

Entity: CPU					
Attribute Name	**Key**	**Type**	**Size**	**Required**	**Validation**
PartID	Primary	Int	11	Yes	Auto increment
PartName		Text	30	YES	
PartCost		Int	11	Yes	
Maker		text	30	Yes	
TDP		Int	11	Yes	
Socket		Text	30	yes	
ClockSpeed		Text	30	yes	
Cores		Text	30	yes	

Entity: GPU					
Attribute Name	**Key**	**Type**	**Size**	**Required**	**Validation**
PartID	Primary	Int	11	Yes	Auto increment
PartName		Text	30	YES	
PartCost		Int	11	Yes	
Maker		text	30	Yes	
TDP		Int	11	Yes	
VRAM		Int	30	yes	
ClockSpeed		Text	30	yes	

Entity: Memory					
Attribute Name	**Key**	**Type**	**Size**	**Required**	**Validation**
PartID	Primary	Int (11)	11	Yes	Auto increment
PartName		Text	30	YES	
PartCost		Int (11)	11	Yes	
Ram		text	30	Yes	
Modules		Text	30	yes	

Entity: Motherboard					
Attribute Name	**Key**	**Type**	**Size**	**Required**	**Validation**
PartID	Primary	Int(11)	11	Yes	Auto increment
PartName		Text	30	YES	
PartCost		Int(11)	11	Yes	
Maker		text	30	Yes	
MaxRam		Int	11	Yes	
Socket		Text	30	yes	

BoardType		Text	30	yes	

Entity: PCcase					
Attribute Name	**Key**	**Type**	**Size**	**Required**	**Validation**
PartID	Primary	Int(11)	11	Yes	Auto increment
PartName		Text	30	YES	
PartCost		Int(11)	11	Yes	
Maker		text	30	Yes	
Boardtype		Text	30	yes	

Entity: PSU					
Attribute Name	**Key**	**Type**	**Size**	**Required**	**Validation**
PartID	Primary	Int(11)	11	Yes	Auto increment
PartName		Text	30	YES	
PartCost		Int(11)	11	Yes	
Maker		text	30	Yes	
Wattage		Int	11	Yes	
Modularity		Text	30	yes	

Entity: USER					
Attribute Name	**Key**	**Type**	**Size**	**Required**	**Validation**
BuildID	Primary	Int(11)	11	Yes	Auto increment
PartName		Text	30	YES	
PartID		int	11		
PartCost		Int(11)	11	Yes	
Build		text	30	Yes	
Type		Text	30	yes	

Query design

Field(s) and Calculation(s)	Select *
Table(s) Query(queries)	CPU
Search criteria	PartName = '".$searchterm1."'
Grouping	
Having	
Sort order	

Implementation

Search.php

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>

<head>
      <title>PC Parts</title>
<link href=”project/main.css” >
  <?php
  echo "<link rel='stylesheet' type='text/css' href='main.css' />";
?>

   </head>
    <header>
   Home
        <nav>
<ul>
<li><a href="ProjectMain.php">Home</a></li>
<li><a href="Info.php">Info</a></li>
<li><a href="Search.php">Build a PC</a></li>
<li><a href="new.php">User builds</a></li>
</ul>
</nav>
   </header>



<body>
```

```
   <form  action="PartPick.php" method="post">
<div class="wrapper">



   proccesor
      <br><select name="Processor">
  <optgroup label="AMD only">
  <option value="Ryzen 5 3600">Ryzen 5 3600</option>
  <option value="Ryzen 3 1200">Ryzen 3 1200</option>
  <option value="Ryzen 7 3800X">Ryzen 7 3800X</option>
  <option value="Ryzen 5 3400G">"Ryzen 5 3400G</option>
      </optgroup>
      <optgroup label = "Intel only">
  <option value="Intel I5 8400">Intel I5 8400</option>
  <option value="Intel I9 9900K">Intel I9 9900K</option>
  <option value="Intel I7 9700K">Intel I7 9700K</option>
    </optgroup>
  </select>









     <br>


   Motherboard
     <br><select name="Board">
<optgroup label="AMD only">
<option value="B450M PRO4">ASRock B450M PRO4 - Micro ATX</option>
<option value="ROG STRIX Z390-I">Asus ROG STRIX Z390-I - ATX</option>
<option value="ROG STRIX B450-F GAMING">ROG STRIX B450-F GAMING - ATX</option>
```

```
      </optgroup>
      <optgroup label = "Intel only">
 <option value="Z390-A PRO">Z390-A PRO - ATX</option>
 <option value="ROG STRIX Z390-E GAMING">ROG STRIX Z390-E GAMING- Micro ATX</option>
 <option value="PRIME Z390-A">PRIME Z390-A - ATX</option>
   </optgroup>
</select>
      <br>


       GPU
      <br><select name="Card">
      <optgroup Label="4GB">
 <option value="GeForce GTX 1050 Ti">GeForce GTX 1050 Ti</option>
 <option value="Radeon RX 570">Radeon RX 570</option>
</optgroup>
 <optgroup Label="8GB">
  <option value="Radeon RX 5700 XT">Radeon RX 5700 XT</option>
<option value="Radeon RX VEGA 56 ">Radeon RX VEGA 56 </option>
<option value="GeForce RTX 2080 SUPER">GeForce RTX 2080 SUPER</option>
</optgroup>
<optgroup label="11GB">
 <option value="GeForce RTX 2080 Ti">GeForce RTX 2080 Ti</option>
        </optgroup>
</select> <br>


       Case
      <br><select name="Case">
      <optgroup Label="ATX">
 <option value="H510 ATX Mid Tower Case">H510 ATX Mid Tower Case</option>
 <option value="P300 ATX Mid Tower Case">P300 ATX Mid Tower Case</option>
```

```
 <option value="PC-O11 Dynamic ATX Full Tower Case">PC-O11 Dynamic ATX Full Tower Case</option>
        <option value="Corsair 750D ATX Full Tower Case">Corsair 750D ATX Full Tower Case</option>
        </optgroup>
      <optgroup Label="Micro-ATX">
      <option value="MasterBox Q300L MicroATX Mini Tower Case">MasterBox Q300L MicroATX Mini Tower
Case</option>
 <option value="MATREXX 30 MicroATX Mini Tower Case">MATREXX 30 MicroATX Mini Tower Case</option>
      </optgroup>
</select>
   <input type="submit">
</form>



<div class="push"></div>
 </div>
    <footer id="footer"> All information sourced from PCPartPicker.com</footer>


   </body>
</html>
```

Info.php

```
<?php
session_start();
?>
<!DOCTYPE html>




<html>


<head>
      <title>PC Parts</title>


<?php
echo "<link rel='stylesheet' type='text/css' href='main.css' />";
?>
   <?php
$username = "root";
$password = "";
$servername = "localhost";
$dbname = "Parts";
$conn = mysqli_connect($servername, $username, $password, $dbname);


?>


   </head>


    <header>
   Info
         <nav>
<ul>
<li><a href="ProjectMain.php">Home</a></li>
```

```
<li><a href="Info.php">Info</a></li>

<li><a href="Search.php">Build a PC</a></li>

<li><a href="new.php">User builds</a></li>

</ul>

</nav>

  </header>



<body>

 <div class="wrapper">

<h1 >CPU <p> CPU (pronounced as separate letters) is the abbreviation for central processing unit. Sometimes referred to simply as the central processor, but more commonly called a processor, the CPU is the brains of the computer where most calculations take place. In terms of computing power, the CPU is the most important element of a computer system.</p></h1>



  <?php

$sql="SELECT PartCost, PartName, Cores, Maker, TDP, ClockSpeed FROM CPU ORDER BY RAND() LIMIT 1";

$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {

  // output data of each row

  while($row = mysqli_fetch_assoc($result)) {



     echo  "<table><tr><th>Cost in £s</th><th>Name</th><th>Cores</th><th>Clock Speed</th><th>Maker</th><th>TDP</th></tr>";

    echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["Cores"].  " </td> <td>" . $row["ClockSpeed"].  " </td><td>". $row["Maker"]. " </td><td>". $row["TDP"].  "</td></tr>";

  }

  echo "</table>";

} else {

  echo "0 results";

}
```

```
   ?>


      <h1>Motherboard

   <p>A motherboard provides connectivity between the hardware components of a computer, like the processor (CPU), memory (RAM), hard drive, and video card. There are multiple types of motherboards, designed to fit different types and sizes of computers.

<br>

Each type of motherboard is designed to work with specific types of processors and memory, so they are not capable of working with every processor and type of memory. However, hard drives are mostly universal and will work with the majority of motherboards, regardless of the type or brand.</p></h1>


       <?php


$sql="SELECT PartCost, PartName, BoardType, MaxRam, Maker, Socket FROM motherboard ORDER BY RAND() LIMIT 1";

$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {

   // output data of each row

    while($row = mysqli_fetch_assoc($result)) {

      echo  "<table><tr><th>Cost in £s</th><th>Name</th><th>Type</th><th>MaxRam</th><th>Maker</th><th>Socket</th></tr>";

     echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["BoardType"]. " </td><td>" . $row["MaxRam"].  " </td><td>". $row["Maker"]. " </td><td>". $row["Socket"].  "</td></tr>";

   }

   echo "</table>";


} else {

   echo "0 results";

}


   ?>

 <h1 >GPU <p> (Graphics Processing Unit) A programmable logic chip (processor) specialized for display functions. The GPU renders images, animations and video for the computer's screen. GPUs are located on plug-in cards, in a chipset on the motherboard or in the same chip as the CPU (see diagram below). See logic chip.
```

A GPU performs parallel operations. Although it is used for 2D data as well as for zooming and panning the screen, a GPU is essential for smooth decoding and rendering of 3D animations and video. The more sophisticated the GPU, the higher the resolution and the faster and smoother the motion in games and movies. GPUs on stand-alone cards include their own memory, while GPUs built into the chipset or CPU chip share main memory with the CPU</p></h1>

```
<?php

$sql="SELECT PartCost, PartName, VRAM, Maker, TDP, ClockSpeed FROM GPU ORDER BY RAND() LIMIT 1";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
  // output data of each row
   while($row = mysqli_fetch_assoc($result)){
     echo  "<table><tr><th>Cost in £s</th><th>Name</th><th>Vram</th><th>Maker</th><th>TDP</th><th>Clock Speed</th></tr>";
     echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["VRAM"].  " </td><td>". $row["Maker"]. " </td><td>". $row["TDP"]."<td>" . $row["ClockSpeed"]. "</td>";
  }
  echo "</table>";
} else {
  echo "0 results";
}

  ?>
   <h1 >PSU <p> A power supply is a hardware component that supplies power to an electrical device. It receives power from an electrical outlet and converts the current from AC (alternating current) to DC (direct current), which is what the computer requires. It also regulates the voltage to an adequate amount, which allows the computer to run smoothly without overheating. The power supply an integral part of any computer and must function correctly for the rest of the components to work.</p></h1>

  <?php

$sql="SELECT PartCost, PartName, modularity, Maker,Wattage FROM psu ORDER BY RAND() LIMIT 1";
$result = mysqli_query($conn, $sql);
```

```
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {


      echo  "<table><tr><th>Cost in
£s</th><th>Name</th><th>modularity</th><th>Maker</th><th>Wattage</th></tr>";
     echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["modularity"].  "
</td><td>". $row["Maker"]. " </td><td>". $row["Wattage"].  "</td></tr>";
   }
   echo "</table>";
} else {
   echo "0 results";
}


   ?>




</div>
<div class="push"></div>
</div>
 <footer id="footer"> All information sourced from PCPartPicker.com</footer>
</div>
</body>
</html>
```

Main.CSS

```
nav li {
   display: inline; padding:50px;
         font: 40px sans-serif;
      margin-Bottom: 10px;

}
nav ul li {




}
html, body {
 height: 100%;
 margin: 0;
}
nav {
   background-color:#AEDFE4; /* Used if the image is unavailable */



}
Header {
      margin-bottom:10px;
       background-color: #AEDFE4;
       font: 30px arial;

}

h1 {
```

```
            font: 30px arial;
}
P{
            font: 18px arial;
   background-size: cover; /* Resize the background image to cover the entire container */}
html {background-color: #C0C0C0}

.info {background-color: #41b3a3;
      font: 18px arial;
   background-size: cover; /* Resize the background image to cover the entire container */}
H3 {
       font: 30px arial
       text-decoration: underline;
}
footer {

       background-color: #AEDFE4;
       font: 30px arial;
position: relative;
right: 0;
bottom: 0;
left: 0;
padding: 1rem;
}
#myDIV {
font: 18px arial;
   display:none;
}
#myDIV1 {
font: 18px arial;
   display:none;
}
```

```
table, th, td {
      border: 1px solid black;
}

   @media print {
div1{ display: none;}
            div2{ display: none;}
            div3{ padding: 30px}
}
@media screen and (max-width: 1080px) {
nav li {
   display: inline; padding:10px;
         font: 30px arial;
      margin-Bottom: 10px;

}

}
@media screen and (max-width: 800px) {
nav li {
   display: inline; padding:10px;
         font: 15px arial;
      margin-Bottom: 10px;

}

}
body{
margin: 0px;
}
.PHP{
```

```
margin-left: 10px;


}


.wrapper {
  min-height: 100%;

  /* Equal to height of footer */
  /* But also accounting for potential margin-bottom of last child */
  margin-bottom: -50px;
}
#footer,
.push {
  height: 50px;
}

.column {
  float: left;
  width: 30%;
}

/* Clear floats after the columns */
.row:after {
  content: "";
  display: table;
  clear: both;
}
.title
{
    font-size: 40px
}
```

PartPick.PHP

```
<?php
session_start();
?>
<!DOCTYPE html>


<html>


<head>
      <title>PC Parts</title>
<link href=”project/main.css” >
  <?php
  echo "<link rel='stylesheet' type='text/css' href='main.css' />";


?>
    <?php
$username = "root";
$password = "";
$servername = "localhost";
$dbname = "Parts";
$conn = mysqli_connect($servername, $username, $password, $dbname);


?>
   </head>
   <div id="page-container">
 <div id="content-wrap">
   <div1>
    <header>
   Home
        <nav>
<ul>
<li><a href="ProjectMain.php">Home</a></li>
```

```
<li><a href="Info.php">Info</a></li>
<li><a href="Search.php">Build a PC</a></li>
<li><a href="new.php">User builds</a></li>
</ul>
</nav>
   </header>
 </div1>
  <div3>
<body>
<div class="wrapper">
   <?php
    $searchterm1 = $_POST["Processor"];?>
   <?php
    $searchterm2 = $_POST["Board"];?>
      <?php
    $searchterm3 = $_POST["Card"];?>
   <?php
    $searchterm4 = $_POST["Case"];?>

 <div class="flip-scale-up-diag-2">



   <h2>Here is a suggested build based on your choices</h2>
   <h3>CPU</h3>

<?php



$sql="SELECT * FROM CPU  WHERE PartName = '".$searchterm1."'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
```

```
   while($row = mysqli_fetch_assoc($result)) {


        echo  "<table><tr><th>Cost in £s</th><th>Name</th><th>Cores</th><th>Clock
Speed</th><th>Maker</th><th>TDP</th></tr>";

      echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["Cores"].  " </td> <td>" .
$row["ClockSpeed"].  " </td><td>". $row["Maker"]. " </td><td>". $row["TDP"].  "</td></tr>";

      $query1 = mysqli_query($conn, "SELECT PartCost FROM CPU /* selects partcost from search and sets Price 1 as
PartCost */

   WHERE PartName = '".$searchterm1."'");

$aName1 = mysqli_fetch_assoc($query1);

$price1  = $aName1['PartCost'];

      $query1 = mysqli_query($conn, "SELECT PartName FROM CPU /* selects PartName from search and sets Partname1
as Partname */

   WHERE PartName = '".$searchterm1."'");

$aName1 = mysqli_fetch_assoc($query1);

$Part1  = $aName1['PartName'];

      $query1 = mysqli_query($conn, "SELECT PartID FROM CPU /* selects PartID from search and sets PartID 1 as partID
*/

   WHERE PartName = '".$searchterm1."'");

$aName1 = mysqli_fetch_assoc($query1);

$ID1  = $aName1['PartID'];

$query1 = mysqli_query($conn, "SELECT TDP FROM CPU /* selects TDP from search and sets TDP 1 as TDP */

WHERE PartName = '".$searchterm1."'");

$aName1 = mysqli_fetch_assoc($query1);

$TDP1  = $aName1['TDP'];

   }


   echo "</table>";



} else {

   echo "0 results";

}
```

```
  ?>



  <br>


  <h3>MotherBoard</h3>

      <?php


$sql="SELECT * FROM motherboard WHERE PartName = '".$searchterm2."'  ORDER BY RAND() LIMIT 1";

$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {

  // output data of each row

  while($row = mysqli_fetch_assoc($result)) {

     echo  "<table><tr><th>Cost</th><th>Name</th><th>Board Size</th><th>Max
Ram</th><th>Maker</th><th>Socket</th></tr>";

    echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["BoardType"]. "</td><td>"
. $row["MaxRam"].  " </td><td>". $row["Maker"]. " </td><td>". $row["Socket"].  "</td></tr>";

          $query2 = mysqli_query($conn, "SELECT PartCost FROM motherboard

   WHERE PartName = '".$searchterm2."'");

$aName2 = mysqli_fetch_assoc($query2);

$price2  = $aName2['PartCost'];

     $query2 = mysqli_query($conn, "SELECT PartName FROM motherboard

   WHERE PartName = '".$searchterm2."'");

$aName2 = mysqli_fetch_assoc($query2);

$Part2  = $aName2['PartName']; $query2 = mysqli_query($conn, "SELECT MaxRam FROM motherboard

   WHERE PartName = '".$searchterm2."'");

$aName2 = mysqli_fetch_assoc($query2);

$Max  = $aName2['MaxRam'];

      $query2 = mysqli_query($conn, "SELECT PartID FROM motherboard

  WHERE PartName = '".$searchterm2."'");

$aName2 = mysqli_fetch_assoc($query2);

$ID2  = $aName2['PartID'];
```

```
    }
    echo "</table>";









} else {
    echo "0 results";
}



    ?>


    <h3>GPU</h3>
     <?php



$sql="SELECT * FROM GPU WHERE PartName = '".$searchterm3."' ORDER BY RAND() LIMIT 1";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    // output data of each row


    while($row = mysqli_fetch_assoc($result)){
        echo
"<table><tr><th>Cost</th><th>Name</th><th>ClockSpeed</th><th>Maker</th><th>TDP</th><th>VRAM</th></tr>";
      echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["Clockspeed"].  "
</td><td>". $row["Maker"]. " </td><td>". $row["TDP"].  "</td><td>". $row["Vram"].  "</td></tr>";
        $query3 = mysqli_query($conn, "SELECT PartCost FROM GPU
     WHERE   PartName = '".$searchterm3."'");
$aName3 = mysqli_fetch_assoc($query3);
$price3  = $aName3['PartCost'];
        $query3 = mysqli_query($conn, "SELECT PartName FROM GPU
```

```
    WHERE PartName = '".$searchterm3."'");
$aName3 = mysqli_fetch_assoc($query3);
$Part3  = $aName3['PartName'];
      $query3 = mysqli_query($conn, "SELECT PartID FROM GPU
   WHERE PartName = '".$searchterm3."'");
$aName3 = mysqli_fetch_assoc($query3);
$ID3  = $aName3['PartID'];
$query3 = mysqli_query($conn, "SELECT TDP FROM GPU
WHERE PartName = '".$searchterm3."'");
$aName3 = mysqli_fetch_assoc($query3);
$TDP2  = $aName3['TDP'];
   }

   echo "</table>";



} else {
   echo "0 results";
}

   ?>
<?php $TTDP = $TDP2 + $TDP1?>
   <h3>PSU</h3>
    <?php



$sql="SELECT * FROM psu where Wattage >= '".$TTDP."' ORDER BY RAND() LIMIT 1";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {
     {
```

```
      echo  "<table><tr><th>Cost</th><th>Name</th><th>Modularity</th><th>Maker</th><th>Wattage</th></tr>";
     echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["Modularity"].  "
</td><td>". $row["Maker"]. " </td><td>". $row["Wattage"].  "</td></tr>";
     $query4 = mysqli_query($conn, "SELECT PartCost FROM psu
    WHERE Partname =  '".$row["PartName"]."'");
$aName4 = mysqli_fetch_assoc($query4);
$price4  = $aName4['PartCost'];
         $query4 = mysqli_query($conn, "SELECT PartName FROM psu
    WHERE Partname =  '".$row["PartName"]."'");
$aName4 = mysqli_fetch_assoc($query4);
$Part4  = $aName4['PartName'];
$query4 = mysqli_query($conn, "SELECT PartID FROM PSU
   WHERE PartName = '".$row["PartName"]."'");
$aName4 = mysqli_fetch_assoc($query4);
$ID4  = $aName4['PartID'];

   }
   echo "</table>";


   }
} else {
   echo "0 results";
}


   ?>
   <H3>PC Case</H3>
   <?php


$sql="SELECT * FROM pccase WHERE PartName = '".$searchterm4."' ORDER BY RAND() LIMIT 1";
```

```
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {
      {
       echo  "<table><tr><th>Cost</th><th>Name</th><th>Size</th><th>Maker</th></tr>";
      echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["BoardType"].  "
</td><td>". $row["Maker"]. " </td></tr>";
     $query5 = mysqli_query($conn, "SELECT PartCost FROM pccase
    WHERE Partname =  '".$row["PartName"]."'");
$aName5 = mysqli_fetch_assoc($query5);
$price6  = $aName5['PartCost'];
          $query5 = mysqli_query($conn, "SELECT PartName FROM pccase
    WHERE Partname =  '".$row["PartName"]."'");
$aName5 = mysqli_fetch_assoc($query5);
$Part6  = $aName5['PartName'];
         $query5 = mysqli_query($conn, "SELECT PartID FROM pccase
   WHERE PartName = '".$searchterm4."'");
$aName5 = mysqli_fetch_assoc($query5);
$ID5  = $aName5['PartID'];

   }
   echo "</table>";


   }
} else {
   echo "0 results";
}


   ?><H3> Ram</H3>
```

```
  <?php



$sql="SELECT * FROM memory WHERE Ram < '".$Max."' ORDER BY RAND() LIMIT 1";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    {
     echo  "<table><tr><th>Cost</th><th>Name</th><th>Ram</th><th>Maker</th><th>Modules</th></tr>";
    echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["Ram"].  " </td><td>".
$row["Maker"]. " </td><td>". $row["Modules"].  "</td></tr>";
    $query6 = mysqli_query($conn, "SELECT PartCost FROM memory
   WHERE Partname =  '".$row["PartName"]."'");
$aName6 = mysqli_fetch_assoc($query6);
$price5  = $aName6['PartCost'];
        $query6 = mysqli_query($conn, "SELECT PartName FROM memory
   WHERE Partname =  '".$row["PartName"]."'");
$aName6 = mysqli_fetch_assoc($query6);
$Part5  = $aName6['PartName'];
         $query6 = mysqli_query($conn, "SELECT PartID FROM memory
   WHERE PartName = '".$row["PartName"]."'");
$aName6 = mysqli_fetch_assoc($query6);
$ID6  = $aName6['PartID'];


  }
  echo "</table>";



  }
} else {
  echo "0 results";
```

```
}



  ?>


  <?php
  $totalprice = $price1 + $price2 + $price3 + $price4 + $price5 + $price6
  ?>
<?php
  $_SESSION["PSU"] =  $Part4;
  $_SESSION["CPU"] =  $Part1;
  $_SESSION["Board"] =  $Part2;
  $_SESSION["GPU"] =  $Part3;
  $_SESSION["RAM"] =  $Part5;
  $_SESSION["Case"] =  $Part6;
  $_SESSION["price1"] =  $price1;
  $_SESSION["price2"] =  $price2;
  $_SESSION["price3"] =  $price3;
  $_SESSION["price4"] =  $price4;
  $_SESSION["price5"] =  $price5;
  $_SESSION["price6"] =  $price6;
  $_SESSION["ID1"] =  $ID1;
  $_SESSION["ID2"] =  $ID2;
  $_SESSION["ID3"] =  $ID3;
  $_SESSION["ID4"] =  $ID4;
  $_SESSION["ID5"] =  $ID5;
  $_SESSION["ID6"] =  $ID6;
  ?>
  <br>


  <?php echo "your total price for this build is " .$totalprice?>
```

```
    <h1>Would you like to save build?</h1>
   <form action="upload.php" method="post">
   BuildName: <input type="text" name="name"><br>
   <input type="submit">
</form>
</div>


   </div3>
  </div>
  <div class="push"></div>
  </div>
<footer id="footer"> All information sourced from PCPartPicker.com</footer>
</body>
</html>
```

Upload.php

```
<?php
session_start();
?>
<!DOCTYPE html>


<html>




<head>
      <title>PC Parts</title>
<link href=”project/main.css” >
  <?php
  echo "<link rel='stylesheet' type='text/css' href='main.css' />";
?>
        <?php
$username = "root";
$password = "";
$servername = "localhost";
$dbname = "Parts";
$conn = mysqli_connect($servername, $username, $password, $dbname);


?>
   </head>
   <div1>
    <header>
   Home




            <nav>
<ul>
<li><a href="ProjectMain.php">Home</a></li>
```

```
<li><a href="Info.php">Info</a></li>
<li><a href="Search.php">Build a PC</a></li>
<li><a href="new.php">User builds</a></li>
</ul>
</nav>
      <?php
echo "The current date is ";
echo date("l F d, Y");
?>
   </header>
   </div1>



<body>
 <div class="wrapper">

 <?php  $_SESSION["BuildName"] = $_POST["name"]; ?>



<?php $Buildname = $_SESSION["BuildName"];?>









<?php
$query = mysqli_query($conn,"SELECT Build FROM User WHERE Build='".$Buildname."'");
if (mysqli_num_rows($query) != 0)
 {
   echo "Name has already been entered";
   header( "refresh:5;url=Exist.php" );
  }
```

```
 else
{

header( "refresh:5;url=Clear.php" );
}




?>
```

```
<div class="push"></div>
 </div>
  <footer id="footer"> All information sourced from PCPartPicker.com</footer>
 </div>
 </body>
</html>
```

Exist.php

```
<?php
session_start();
?>
<!DOCTYPE html>


<html>
 <head>
      <title>PC Parts</title>
<link href=”project/main.css” >
  <?php
  echo "<link rel='stylesheet' type='text/css' href='main.css' />";
?>
       <?php
$username = "root";
$password = "";
$servername = "localhost";
$dbname = "Parts";
$conn = mysqli_connect($servername, $username, $password, $dbname);


?>
   </head>
   <div1>
    <header>
   Home



           <nav>
<ul>
<li><a href="ProjectMain.php">Home</a></li>
<li><a href="Info.php">Info</a></li>
<li><a href="Search.php">Build a PC</a></li>
```

```
<li><a href="new.php">User builds</a></li>
</ul>
</nav>
      <?php
echo "The current date is ";
echo date("l F d, Y");
?>
   </header>
   </div1>



<body>
 <div class="wrapper">
 <h1>Please enter a new name for your build</h1>
<form action="upload.php" method="post">
BuildName: <input type="text" name="name"><br>
<input type="submit">
</form>









<div class="push"></div>
```

```
</div>
   <footer id="footer"> All information sourced from PCPartPicker.com


   </body>
   </footer>
</html>
```

Clear.php

```
<?php
session_start();
?>
<!DOCTYPE html>


<html>



<head>
      <title>PC Parts</title>
<link href="project/main.css" >

  <?php
  echo "<link rel='stylesheet' type='text/css' href='main.css' />";
?>
        <?php
$username = "root";
$password = "";
$servername = "localhost";
$dbname = "Parts";
$conn = mysqli_connect($servername, $username, $password, $dbname);


?>
   </head>
   <div1>
    <header>
   Home



            <nav>
<ul>
```

```
<li><a href="ProjectMain.php">Home</a></li>
<li><a href="Info.php">Info</a></li>
<li><a href="Search.php">Build a PC</a></li>
<li><a href="new.php">User builds</a></li>
</ul>
</nav>
      <?php
echo "The current date is ";
echo date("l F d, Y");
?>
  </header>
  </div1>



<body>
<div class="wrapper">
   <?php $CPU = $_SESSION["CPU"] ;?>

<?php $Price1 = $_SESSION["price1"] ;?>
  <?php $ID1 = $_SESSION["ID1"] ;?>

<?php $Board = $_SESSION["Board"] ;?>

<?php $Price2 = $_SESSION["price2"] ;?>
  <?php $ID2 = $_SESSION["ID2"] ;?>

<?php $GPU = $_SESSION["GPU"] ;?>

<?php $Price3 = $_SESSION["price3"] ;?>
 <?php $ID3 = $_SESSION["ID3"] ;?>
<?php $PSU = $_SESSION["PSU"] ;?>
```

```
<?php $Price4 = $_SESSION["price4"] ;?>
   <?php $ID4 = $_SESSION["ID4"] ;?>
   <?php $RAM = $_SESSION["RAM"] ;?>

<?php $Price5 = $_SESSION["price5"] ;?>
   <?php $ID5 = $_SESSION["ID5"] ;?>
<?php $CASE = $_SESSION["Case"] ;?>

<?php $Price6 = $_SESSION["price6"] ;?>
   <?php $ID6 = $_SESSION["ID6"] ;?>

<?php $Buildname = $_SESSION["BuildName"];?>




<?php

$sql =  "INSERT INTO user (Build,PartName,PartCost,PartID,Type) VALUES ('$Buildname','$CPU','$Price1','$ID1','CPU')";
 mysqli_query($conn, $sql);



?>

<?php

$sql =  "INSERT INTO user (Build,PartName,PartCost,PartID,Type) VALUES
('$Buildname','$Board','$Price2','$ID2','MOBA')";
 mysqli_query($conn, $sql);



?>
```

```
<?php



 $sql =  "INSERT INTO user (Build,PartName,PartCost,PartID,Type) VALUES ('$Buildname','$GPU','$Price3','$ID3','GPU')";
mysqli_query($conn, $sql);

?>

<?php



$sql =  "INSERT INTO user (Build,PartName,PartCost,PartID,Type) VALUES ('$Buildname','$PSU','$Price4','$ID4','PSU')";



mysqli_query($conn, $sql);
?>

<?php

$sql =  "INSERT INTO user (Build,PartName,PartCost,PartID,Type) VALUES ('$Buildname','$RAM','$Price5','$ID5','RAM')";
mysqli_query($conn, $sql);

?>




 <?php

$sql =  "INSERT INTO user (Build,PartName,PartCost,PartID,Type) VALUES ('$Buildname','$CASE','$Price6','$ID6','Case')";

mysqli_query($conn, $sql);
```

```
 ?>




<?php header( "refresh:6;url=new.php" ); ?>




<div class="push"></div>
</div>
   <footer id="footer"> All information sourced from PCPartPicker.com



   </footer>
 </body>
</html>
```

PriceCheck.php

```
<?php
session_start();
?>
<!DOCTYPE html>


<html>




<head>
      <title>PC Parts</title>
<link href=="project/main.css"  >
  <?php
  echo "<link rel='stylesheet' type='text/css' href='main.css' />";
?>
     <?php
$username = "root";
$password = "";
$servername = "localhost";
$dbname = "Parts";
$conn = mysqli_connect($servername, $username, $password, $dbname);


?>


   </head>




    <header>
   Price Check
```

```
      <nav>
<ul>
<li><a href="ProjectMain.php">Home</a></li>
<li><a href="Info.php">Info</a></li>
<li><a href="Search.php">Build a PC</a></li>
<li><a href="new.php">User builds</a></li>
</ul>

<?php
echo "The current date is ";
echo date("l F d, Y");
?>
</nav>
   </header>



<body>
 <div class="wrapper">

<?php  $Build = $_POST["name"];
    ?>

<div21 style="display: none">
<?Php



$sql="SELECT PartName FROM user WHERE Build = '".$Build."' and type = 'CPU'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {
```

```
    {
     echo  "<table><tr><th>PartName</th></tr>";
    echo "<tr><td>" . $row["PartName"]. "</td></tr>";
        $query1 = mysqli_query($conn, "SELECT Partname FROM User
WHERE Partname = '".$row["PartName"]."'");
$aName1 = mysqli_fetch_assoc($query1);
$CPUu   = $aName1['Partname'];


}
echo "</table>";



}
} else {
echo "0 results";
} ?>

<?Php



$sql="SELECT PartName FROM user WHERE Build = '".$Build."' and type = 'MOBA'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {
     {
      echo  "<table><tr><th>PartName</th></tr>";
     echo "<tr><td>" . $row["PartName"]. "</td></tr>";
        $query2 = mysqli_query($conn, "SELECT Partname FROM User
WHERE Partname = '".$row["PartName"]."'");
$aName2 = mysqli_fetch_assoc($query2);
$MOBAu  = $aName2['Partname'];
```

```
}
echo "</table>";



}
} else {
echo "0 results";
} ?>

<?Php



$sql="SELECT PartName FROM user WHERE Build = '".$Build."' and type = 'GPU'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {
     {
      echo  "<table><tr><th>PartName</th></tr>";
     echo "<tr><td>" . $row["PartName"]. "</td></tr>";
          $query3 = mysqli_query($conn, "SELECT Partname FROM User
WHERE Partname = '".$row["PartName"]."'");
$aName3 = mysqli_fetch_assoc($query3);
$GPUu  = $aName3['Partname'];


}
echo "</table>";



}
} else {
```

```
echo "0 results";
} ?>



<?Php



$sql="SELECT PartName FROM user WHERE Build = '".$Build."' and type = 'PSU'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {
      {
       echo  "<table><tr><th>PartName</th></tr>";
      echo "<tr><td>" . $row["PartName"]. "</td></tr>";
           $query4 = mysqli_query($conn, "SELECT Partname FROM User
WHERE Partname = '".$row["PartName"]."'");
$aName4 = mysqli_fetch_assoc($query4);
$PSUu  = $aName4['Partname'];


}
echo "</table>";



}
} else {
echo "0 results";
} ?>



<?Php
```

```
$sql="SELECT PartName FROM user WHERE Build = '".$Build."' and type = 'RAM'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {
     {
      echo  "<table><tr><th>PartName</th></tr>";
     echo "<tr><td>" . $row["PartName"]. "</td></tr>";
         $query5 = mysqli_query($conn, "SELECT Partname FROM User
WHERE Partname = '".$row["PartName"]."'");
$aName5 = mysqli_fetch_assoc($query5);
$RAMu  = $aName5['Partname'];

}
echo "</table>";



}
} else {
echo "0 results";
} ?>
<?Php



$sql="SELECT PartName FROM user WHERE Build = '".$Build."' and type = 'Case'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {
     {
      echo  "<table><tr><th>PartName</th></tr>";
```

```
      echo "<tr><td>" . $row["PartName"]. "</td></tr>";
          $query6 = mysqli_query($conn, "SELECT Partname FROM User
WHERE Partname = '".$row["PartName"]."'");
$aName6 = mysqli_fetch_assoc($query6);
$CASEu  = $aName6['Partname'];


}
echo "</table>";



}
} else {
echo "0 results";
} ?>
</div21>


<?php
   $_SESSION["PSU"] =  $PSUu;
   $_SESSION["CPU"] =  $CPUu;
   $_SESSION["Board"] =  $MOBAu;
   $_SESSION["GPU"] =  $GPUu;
   $_SESSION["RAM"] =  $RAMu;
   $_SESSION["Case"] =  $CASEu;
$_SESSION["Build"] = $Build;
 header( "refresh:3;url=display.php" );
   ?>


 </div>
 <div class="push"></div>
 </div>
    <footer id="footer"> All information sourced from PCPartPicker.com</footer>
  </div>
```

```
  </body>



</html>
```

Display.php

```
<?php
session_start();
?>
<!DOCTYPE html>


<html>




<head>
      <title>PC Parts</title>
<link href="project/main.css" >
  <?php
  echo "<link rel='stylesheet' type='text/css' href='main.css' />";
?>
     <?php
$username = "root";
$password = "";
$servername = "localhost";
$dbname = "Parts";
$conn = mysqli_connect($servername, $username, $password, $dbname);


?>
<div1>
   </head>
    <header>
USER BUILD


      <nav>
<ul>
<li><a href="ProjectMain.php">Home</a></li>
```

```
<li><a href="Info.php">Info</a></li>
<li><a href="Search.php">Build a PC</a></li>
<li><a href="new.php">User builds</a></li> /y///
<?php
echo "The current date is ";
echo date("l F d, Y");
?>
</nav>
   </header>
</div1>

<body>
<div class="wrapper">
<?php
    $PSUu =  $_SESSION["PSU"];
    $CPUu  = $_SESSION["CPU"];
    $MOBAu = $_SESSION["Board"];
    $GPUu  = $_SESSION["GPU"];
    $RAMu =  $_SESSION["RAM"] ;
   $Build = $_SESSION["Build"];
    $CASEu = $_SESSION["Case"];



   ?>
<div class = "title">
<?php echo $Build;?>
</div>
<H1 class="roll-in-left">CPU</H1>

<div class = "PHP" >
<?php
```

```
$sql="SELECT * FROM cpu  WHERE PartName ='".$CPUu."'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
   // output data of each row
   while($row = mysqli_fetch_assoc($result)) {

       echo  "<table><tr><th>Cost in £s</th><th>Name</th><th>Cores</th><th>Clock
Speed</th><th>Maker</th><th>TDP</th></tr>";
     echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["Cores"].  " </td> <td>" .
$row["ClockSpeed"].  " </td><td>". $row["Maker"]. " </td><td>". $row["TDP"].  "</td></tr>";

   }

   echo "</table>";

} else {
   echo "0 results";
}

   ?>
</div>
   <br>
   <h1 class ="PHP">MotherBoard</h1>
<div class = "PHP">
   <?php

   $sql="SELECT * FROM motherboard  WHERE PartName  ='".$MOBAu."'";
   $result = mysqli_query($conn, $sql);
   if (mysqli_num_rows($result) > 0) {
```

```
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {

      echo  "<table><tr><th>Cost</th><th>Name</th><th>Max Ram</th><th>Maker</th><th>Socket</th></tr>";
     echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["MaxRam"].  "
</td><td>". $row["Maker"]. " </td><td>". $row["Socket"].  "</td></tr>";

    }

    echo "</table>";

  } else {
    echo "0 results";
  }

    ?>
   </div>
<br>
    <h1>GPU</H1>
     <div class = "PHP">
    <?php

    $sql="SELECT * FROM gpu  WHERE PartName ='".$GPUu."'";
    $result = mysqli_query($conn, $sql);
    if (mysqli_num_rows($result) > 0) {
      // output data of each row
      while($row = mysqli_fetch_assoc($result)) {

        echo
"<table><tr><th>Cost</th><th>Name</th><th>ClockSpeed</th><th>Maker</th><th>TDP</th><th>VRAM</th></tr>";
```

```
        echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["Clockspeed"].  "
</td><td>". $row["Maker"]. " </td><td>". $row["TDP"].  "</td><td>". $row["Vram"].  "</td></tr>";


        }


        echo "</table>";



    } else {
        echo "0 results";
    }


        ?>
      </div>
        <br>
<h1>PSU</h1>
<div class = "PHP">
        <?php



        $sql="SELECT * FROM psu  WHERE PartName ='".$PSUu."'";
        $result = mysqli_query($conn, $sql);
        if (mysqli_num_rows($result) > 0) {
            // output data of each row
            while($row = mysqli_fetch_assoc($result)) {


             echo
"<table><tr><th>Cost</th><th>Name</th><th>Modularity</th><th>Maker</th><th>Wattage</th></tr>";
             echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>" . $row["Modularity"].  "
</td><td>". $row["Maker"]. " </td><td>". $row["Wattage"].  "</td></tr>";;


            }
```

```
            echo "</table>";



        } else {
            echo "0 results";
        }


            ?>
          </div>
<br>
<h1>RAM</h1>
<div class = "PHP">
            <?php



            $sql="SELECT * FROM memory  WHERE PartName ='".$RAMu."'";
            $result = mysqli_query($conn, $sql);
            if (mysqli_num_rows($result) > 0) {
                // output data of each row
                while($row = mysqli_fetch_assoc($result)) {


                  echo
"<table><tr><th>Cost</th><th>Name</th><th>Modularity</th><th>Maker</th><th>Wattage</th></tr>";
                  echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>". $row["Maker"]. "
</td><td>". $row["Ram"].  "</td><td>". $row["Modules"].  "</td></tr>";;


                }


                echo "</table>";



            } else {
                echo "0 results";
```

```
        }


          ?>
         </div>
          <br>
          <h1>Case</h1>
          <div class = "php">


          <?php



          $sql="SELECT * FROM pccase  WHERE PartName ='".$CASEu."'";
          $result = mysqli_query($conn, $sql);
          if (mysqli_num_rows($result) > 0) {
             // output data of each row
             while($row = mysqli_fetch_assoc($result)) {


               echo  "<table><tr><th>Cost</th><th>Name</th><th>Boardtype</th><th>Maker</th></tr>";
              echo "<tr><td>" . $row["PartCost"]. "</td><td>" . $row["PartName"]. "</td><td>". $row["BoardType"]. "
</td><td>". $row["Maker"].  "</td></tr>";;


             }


             echo "</table>";



          } else {
             echo "0 results";
          }


             ?>
</div>
```

```
   <div class="push"></div>
</div>
<div2>
   <footer id="footer"> All information sourced from PCPartPicker.com</footer>
</div3>
  </body>
</html>
```

Research and development of new skills and/or knowledge

The advancement of one web page to another after a certain amount of time using header("refresh:5;url=Exist.php");

It is used to redirect pages that are used for inserting sql statements onto pages that show the tables created from the inserts.

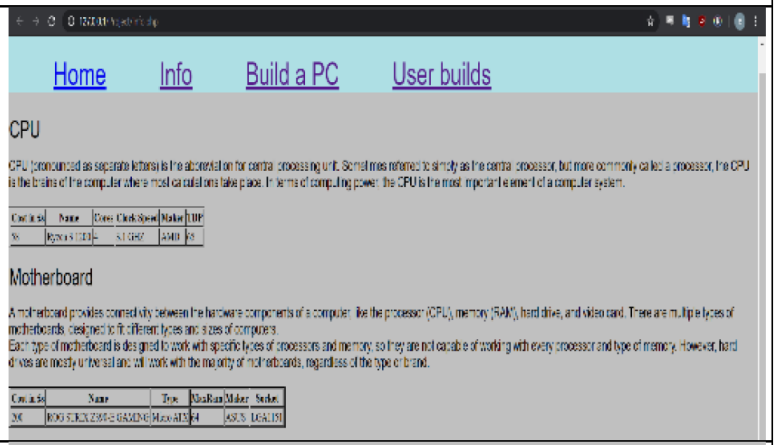
Source: Stackoverflow

Ongoing testing

What you are testing	issues	How issue resolved	References used to resolve
Input validation of submission page	Redirects to exist.php which is used to re-enter build name regardless of input.	The addition of 'if (mysqli_num_rows($query) != 0)'	stackoverflow

Bug	Problem	Fix
PC Case 0 results Ram Cost | Name | Ram | Maker | Modules 163 | Corsair Vengeance RGB Pro | 32 | Corsair | 2 x 16 GB Notice: Undefined variable: price6 in C:\Program Files (x86)\EasyPHP-Devserver-17\eds-www\Project\PartPick.php on line 281 Notice: Undefined variable: Part6 in C:\Program Files (x86)\EasyPHP-Devserver-17\eds-www\Project\PartPick.php on line 288 Notice: Undefined variable: price6 in C:\Program Files (x86)\EasyPHP-Devserver-17\eds-www\Project\PartPick.php on line 294 Notice: Undefined variable: ID5 in C:\Program Files (x86)\EasyPHP-Devserver-17\eds-www\Project\PartPick.php on line 299	Certain database fields don't work	Extra whitespace at end of table data.
Notice: Undefined index: Build in C:\Program Files (x86)\EasyPHP-Devserver-17\eds-www\Project\Display.php on line 54	Build is set as an undefined index	$_SESSION must be in all caps in order to work

Testing

Test Case ID	Test case objective	Test case description	Expected result	Actual result
1	Check Communication between website and database	Enter info page and view PHP Code showing information from database	Information from database shows	
2	Check that a user can enter a new entry into the database	Make a pc Build titled "Test" and submit to database using build name	User submitted page will show a new entry	

Build	PartName	PartCost
Test	Ryzen 3 1200	58
Test	ROG STRIX Z390-I	112
Test	GeForce RTX 2080 Ti	1080
Test	RMx (2018) 80+ Gold Certified	115
Test	G.Skill Ripjaws V	73
Test	PC-O11 Dynamic ATX Full Tower Case	130

3	That a user build can be selected and displayed with all part information	Select Build name from submissions	Build will be displayed with all information	Test CPU Cost in £s | Name | Cores | Clock Speed | Maker | TDP 58 | Ryzen 3 1200 | 4 | 3.1 GHZ | AMD | 65 MotherBoard Cost | Name | Max Ram | Maker | Socket 112 | ROG STRIX Z390-I | 64 | ASUS | AM4 GPU Cost | Name | ClockSpeed | Maker | TDP | VRAM 1080 | GeForce RTX 2080 Ti | 1.5 GHZ | EVGA | 250 | 11 PSU Cost | Name | Modularity | Maker | Wattage 115 | RMx (2018) 80+ Gold Certified | Full | Corsair | 650 RAM Cost | Name | Modularity | Maker | Wattage 73 | G.Skill Ripjaws V | G.Skill | 16 | 2 x 8 GB Case Cost | Name | Boardtype | Maker 130 | PC-O11 Dynamic ATX Full Tower Case | ATX | Lian Li	

4	Not allow existing name to be entered	Will enter Name “test” which is an existing entry in database	A echo will display stating that name already exists and will ask user to enter a different name.		
5	Redirection to enter new name page after entering existing name				

<table>
<tr>
<td>6</td>
<td>Don’t make choices on selection page</td>
<td>Use default values on selection page to</td>
<td>A build having a ryzen 5 3600
Asus B450M Pro4
GTX 1050 TI
H510 Mid Tower
will display</td>
<td>
Here is a suggested build based on your choices

CPU
<table>
<tr><th>Cost in £s</th><th>Name</th><th>Cores</th><th>Clock Speed</th><th>Maker</th><th>TDP</th></tr>
<tr><td>193</td><td>Ryzen 5 3600</td><td>6</td><td>3.6 GHZ</td><td>AMD</td><td>65</td></tr>
</table>

MotherBoard
<table>
<tr><th>Cost</th><th>Name</th><th>Board Size</th><th>Max Ram</th><th>Maker</th><th>Socket</th></tr>
<tr><td>100</td><td>B450M PRO4</td><td>Micro ATX</td><td>64</td><td>ASUS</td><td>AM4</td></tr>
</table>

GPU
<table>
<tr><th>Cost</th><th>Name</th><th>ClockSpeed</th><th>Maker</th><th>TDP</th><th>VRAM</th></tr>
<tr><td>126</td><td>GeForce GTX 1050 Ti</td><td>1.3 GHZ</td><td>MSI</td><td>75</td><td>4</td></tr>
</table>

PSU
<table>
<tr><th>Cost</th><th>Name</th><th>Modularity</th><th>Maker</th><th>Wattage</th></tr>
<tr><td>130</td><td>FOCUS Plus 80+ Gold Certified</td><td>Full</td><td>Seasonic</td><td>850</td></tr>
</table>

PC Case
<table>
<tr><th>Cost</th><th>Name</th><th>Size</th><th>Maker</th></tr>
<tr><td>70</td><td>H510 ATX Mid Tower Case</td><td>ATX</td><td>NZXT</td></tr>
</table>

Ram
<table>
<tr><th>Cost</th><th>Name</th><th>Ram</th><th>Maker</th><th>Modules</th></tr>
<tr><td>75</td><td>Corsair Vengeance LPX</td><td>16</td><td>Corsair</td><td>2 x 8 GB</td></tr>
</table>

your total price for this build is 694
</td>
</tr>
</table>

Evaluation

Fitness for purpose

I set out to build a dynamic website for building PCs that interacts with a database. In my solution, I have used a database to store information about PC parts and I used PHP code to query the database to find parts that matched the user selections for CPU, motherboard, GPU and PC case. I have used the results of queries to add the selected component prices together to get a total price for the build. The user can then choose to save details of their selections. They then need to give a build name that hasn't already been used and the build details are saved to the database. This means that the user is able view details of their build later by entering the build name and having details of all the components displayed on the screen in a table.

Since all of this works as I predicted, I believe that my solution is fit for purpose. I had hoped to add in some JavaScript so that the selection page would have a dynamic selection available. However, I didn't have time to get this part working correctly and I decided to take it out. Since the use of JavaScript goes beyond what is required for the Advanced Higher course, I don't think that removing it affects the fitness for purpose of my solution.

Results of testing

My testing of the website shows that it is fully-working and interacts with the database. Users can view the Information page, select parts for their PC build, view details of their build on a separate page and save the build to the database. If users don't make selections, the default values are used in their build. Mt tests show that the code works and all bugs have been sorted.

I asked a friend to use the website to build a PC and he was able to select the parts he wanted and save the build. When he first tried to save his build, he was told that the build name he used already existed and he had to use a different build name.

Maintainability of code

The external CSS file applied to every page of my website has media queries to control the layout of the site on laptop and mobile phones. If someone needed to adapt the website to display on a tablet device, one extra media query could be added to the CSS file and this would then be available for every page on the website. This type of adaptive maintenance would be easy to achieve.

This separate page would have the database connection code and the code needed to display the menu bar since this code is the same on every page. I would then need to use the PHP include statement to make sure that all the code on that page was included on the other pages of the website. Doing this would make the code for each page shorter and it would then be easier for someone else to do some corrective or perfective maintenance of the code since there would be less of it.

Robustness of code

My code is somewhat robust because I have used validation to make sure that the same build name cannot be used more than once. When asked for a build name, the name entered is checked to see if it already exists. If it does, the user is asked to re-enter the name on a different web page. To make this happen I learned about the PHP header statement and used this to redirect the user as seen in my code. I had wanted to add validation to make sure that special characters and numbers couldn't be used in the build name but I ran out of time to do this part of the work.

Candidate 2 evidence

Advanced Higher Computing Project

Analysis

Description of the problem

My project consists of a program that links to a database of golfers. I will use bubble sort and binary search to sort through these golfers and search for a specific player.

The statistics of golfers will be read into the program from an Access database.

Golfers will then be ordered by average points per tournament in ascending order. This will then be displayed in a list box on the user interface.

Users will be able to use a binary search to search for a player via their player ID that a user has input. The values outputted are their name, age and their number of wins for the year and number of events played in that year.

I will then simulate a matchplay knockout tournament using probability. Golfers will go head to head against each other using probability calculated from their average points. 16 randomly generated golfers will make up this tournament. The competition progress and results of each round will be displayed in text boxes and the winner of the tournament will also be shown in a message box.

There will also be a long drive competition that will also use probability. The long drive will be based on the players' shortest and longest carry distance for that year. 5 drives will be hit by each player and the winner is the single longest drive. The longest attempt for each player and their names will be displayed in a list box.

https://www.kaggle.com/grantruedy/pga-tour-golf-data-2017-season

Scope

This involves creating a modular program. The deliverables include:

- A detailed design of the program structure
- A test plan with a completed test plan data table
- A working program
- The results of testing
- An evaluation report

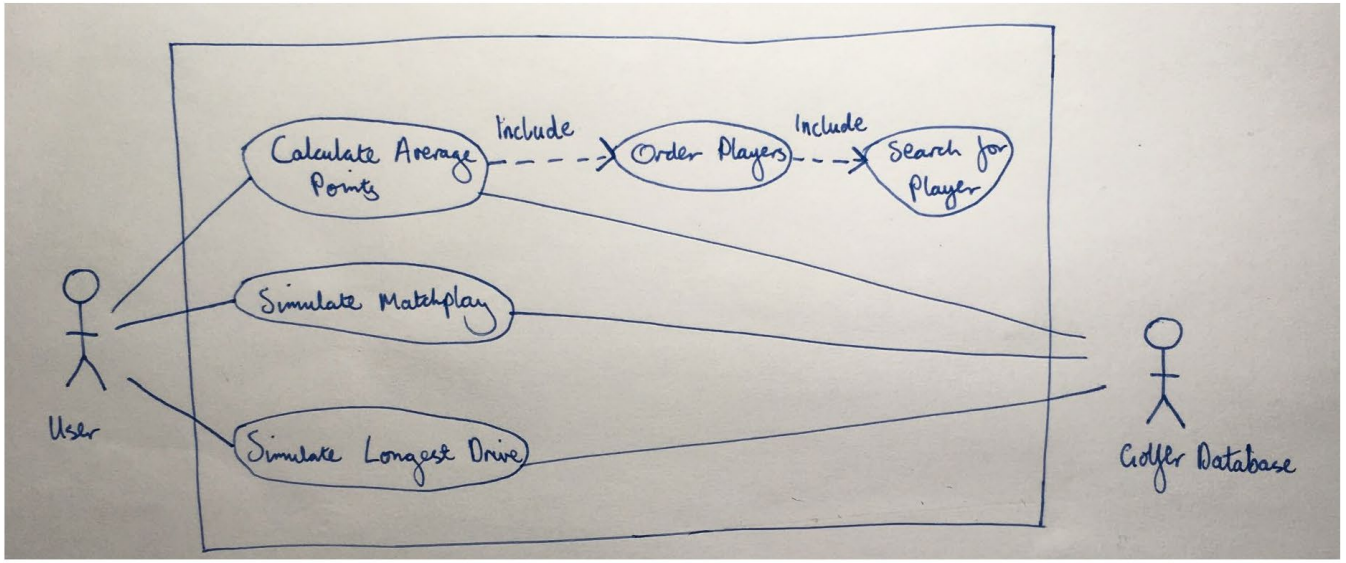
Constraints

- The program must be created using visual basic
- The work must be completed by March 23rd
- The working database and program will run on the windows operating system

Boundaries

- The program will read the golfer's data from an Access database
- The data is accurate so there is no need for input validation at the reading in stage

UML use case diagram



Requirements Specification

End User Requirements

- Read in from a database all players and their statistics from the 2017 PGA Tour season
- Order players' in order of their average points per event
- Search for players by their ID and find their name, age, number of wins and number of events played in the 2017 season
- Simulate a 16-player matchplay tournament
- Simulate a long drive competition between two players

Functional Requirements

Functional requirements are defined in terms of the inputs, processes, and outputs listed below. All inputs are imported from an Access database and outputs are displayed on screen. Each process should be a separate procedure or function that is called from the main program.

Inputs

- External Access database which contains information that can be stored in variables PlayerID, name, eventsPlayed, points, numberOfWins, shortCarry, longCarry and age. An array of records will be used to store this information.
- The user will be asked for a player ID. This ID will be used to search for the corresponding player's details and display them in a message box.
- Two players' names will be input by the user to go head to head in the long drive competition.

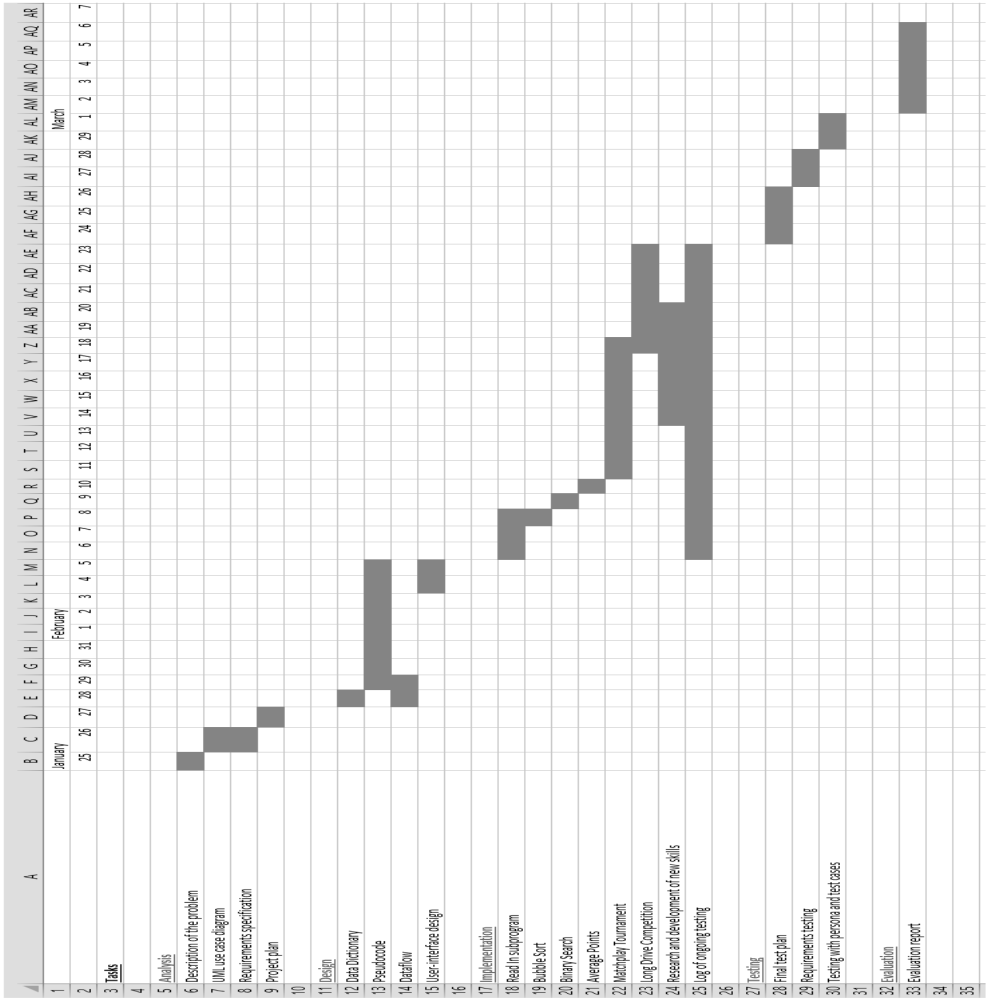
Processes

- Read in the information from an external Access database and store the information in arrays of records.
- Code a binary search algorithm to find a specific playerID that is searched for.
- Calculate the average points per tournament by dividing totalPoints by noEventsPlayed and store this in the array of records alongside the player's other details.
- Code a bubble sort to re-order the list in ascending order of average points.
- Generate 16 random numbers to get 16 random players to form a knockout tournament.
- Ensure that none of the numbers are the same by using validation.
- Use average points to calculate the probability of the player winning the match and use this to determine a winner for that specific match. Call the winner subprogram for every individual match in the tournament to eventually end up with a winner of the whole tournament.
- Validate the two players selected by the user by using input validation for the long dive competition. Ensure the IDs entered are between 1 and 195 and are not the same as each other.
- Generate 5 distances between their longest and shortest carries for each player and select the largest one to compare with the other player's longest drive to determine the winner.

Outputs

- The playerID searched for will be display a message box with their name, age, numberOfWins and EventsPlayed.
- The new ordered list will be displayed in a list box.
- Display the tournament players onscreen in the format of a knockout tournament in textboxes.
- Display the winner of the tournament in a message box.
- Display the two competitors in a message box before the long drive competition is simulated. Add their name and longest drive to a list box once simulated. Display the result in a message box.

Project plan

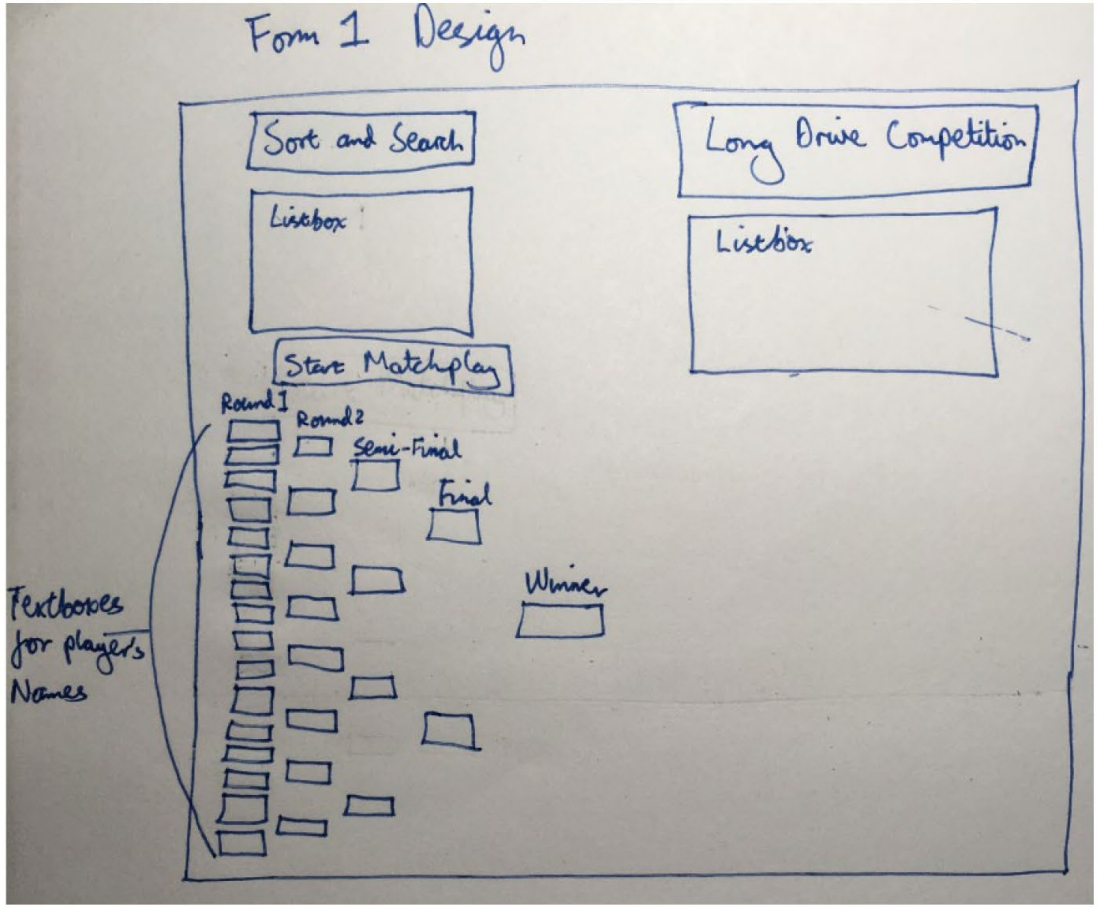


Task	Resources Required
Analysis • **Description of the problem** • **UML use case diagram** • **Requirements specification** • **Project plan**	 • Microsoft Word, Google Chrome, Microsoft One Note • Microsoft Word, Paper • Microsoft Word, Google Chrome • Microsoft Word, Microsoft Excel, Microsoft One Note
Design • **Data Dictionary** • **Pseudocode** • **Dataflow** • **User-interface design**	 • Paper, Microsoft Word • Paper, Microsoft Word • Paper, Microsoft Word • Paper, Microsoft Word
Implementation • **Read In subprogram** • **Bubble sort** • **Binary Search** • **Average points** • **Matchplay Tournament** • **Long Drive Competition** • **Research and development of new skills and/or knowledge** • **Log of ongoing testing**	 • Microsoft Word, Google Chrome, Visual Basic, Microsoft One Note • Microsoft Word, Google Chrome, Visual Basic, Microsoft One Note • Microsoft Word, Google Chrome, Visual Basic, Microsoft One Note • Microsoft Word, Google Chrome, Visual Basic, Microsoft One Note • Microsoft Word, Google Chrome, Visual Basic, Microsoft One Note • Microsoft Word, Google Chrome, Visual Basic, Microsoft One Note

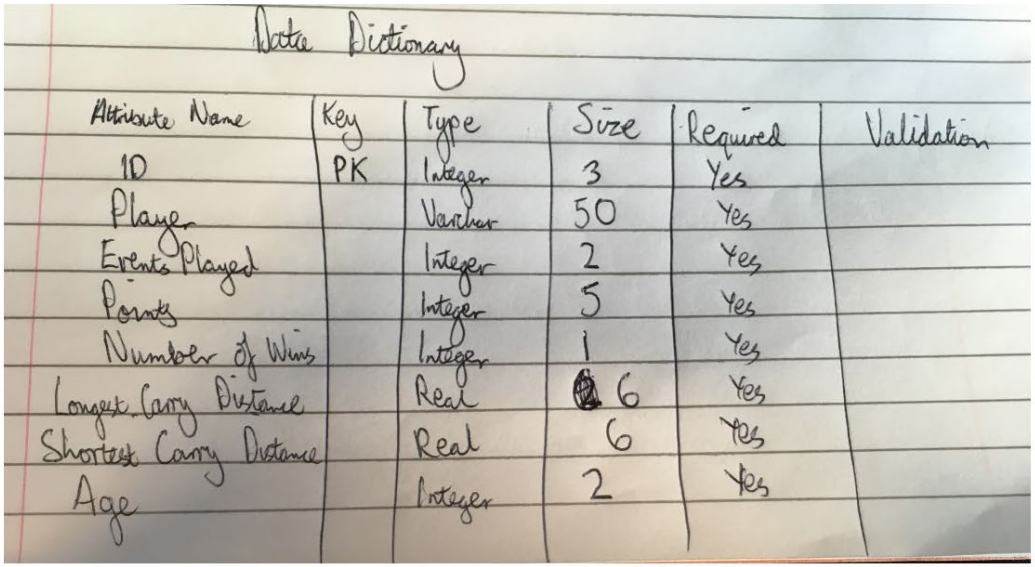
	• Microsoft Word, Google Chrome, Visual Basic • Visual Basic, Microsoft Word
Testing • **Final test plan** • **Requirements testing** • **Testing with persona and test cases**	 • Microsoft Word • Visual Basic, Word, Snipping Tool • Microsoft Word
Evaluation • **Evaluation report**	 • Microsoft Word

Design

Interface Design



Database Design

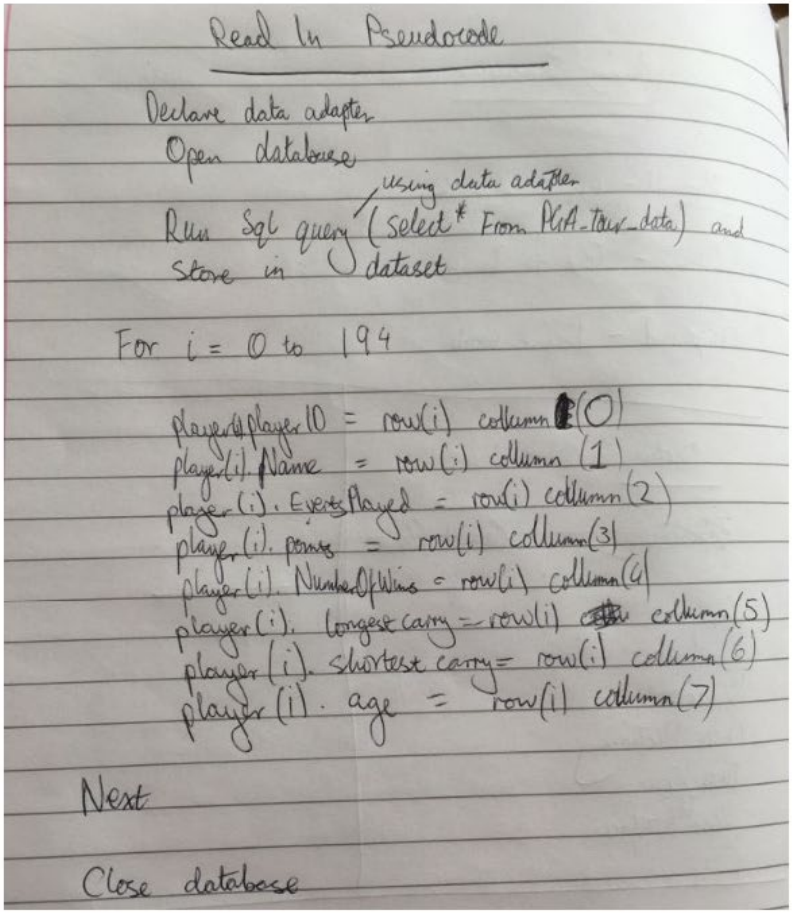


Data Dictionary

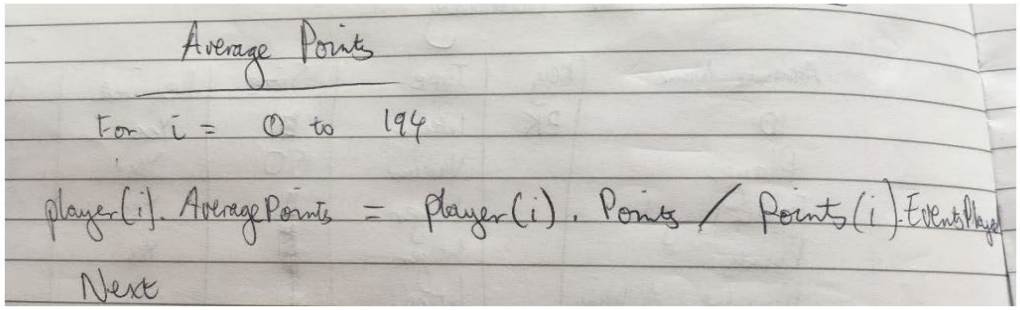
Attribute Name	Key	Type	Size	Required	Validation
ID	PK	Integer	3	Yes	
Player		Varchar	50	Yes	
Events Played		Integer	2	Yes	
Points		Integer	5	Yes	
Number of Wins		Integer	1	Yes	
Longest Carry Distance		Real	6	Yes	
Shortest Carry Distance		Real	6	Yes	
Age		Integer	2	Yes	

Pseudocode

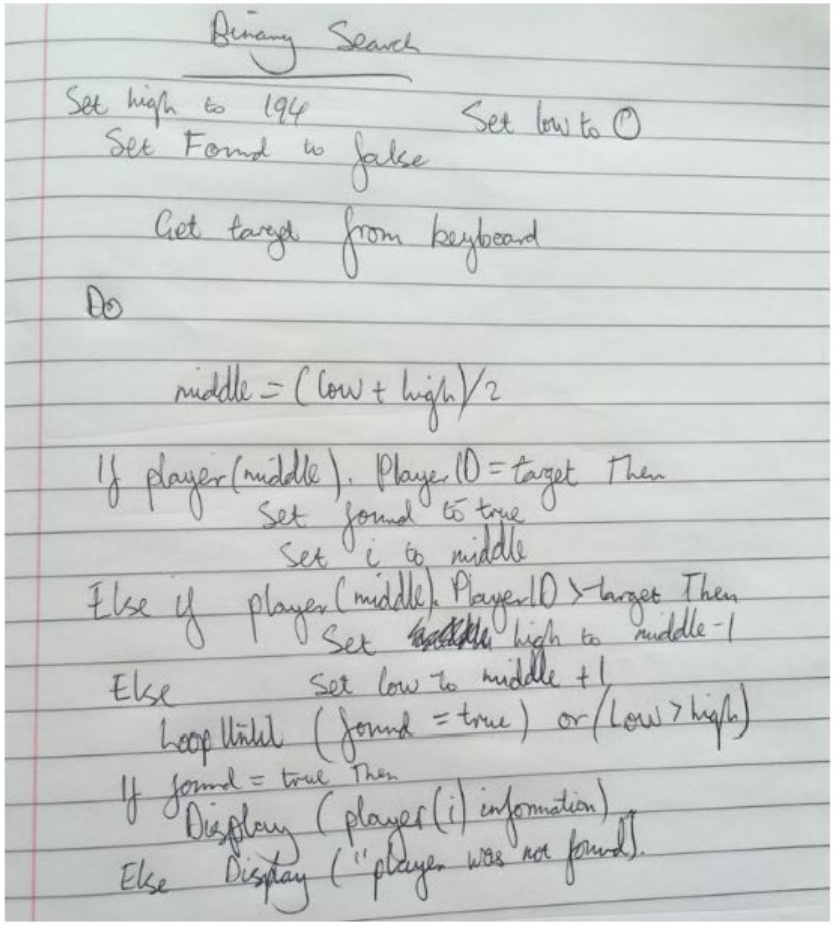
ReadIn



AveragePoints



Binary Search



Binary Search

Set high to 194 Set low to 0

Set Found to false

Get target from keyboard

Do

middle = (low + high)/2

If player(middle).PlayerID = target Then

Set found to true

Set i to middle

Else if player(middle).PlayerID > target Then

Set high to middle - 1

Else Set low to middle + 1

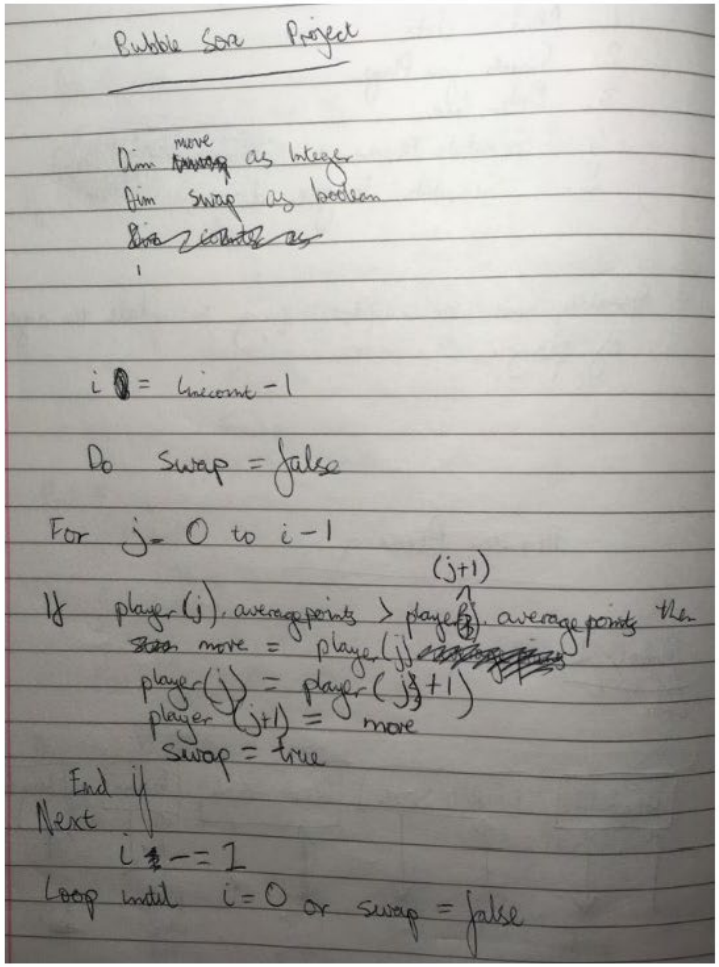
Loop Until (found = true) or (Low > high)

If found = true Then

Display (player(i) information)

Else Display ("player was not found").

Bubble Sort



Bubble Sort Project

Dim move as Integer
Dim swap as boolean

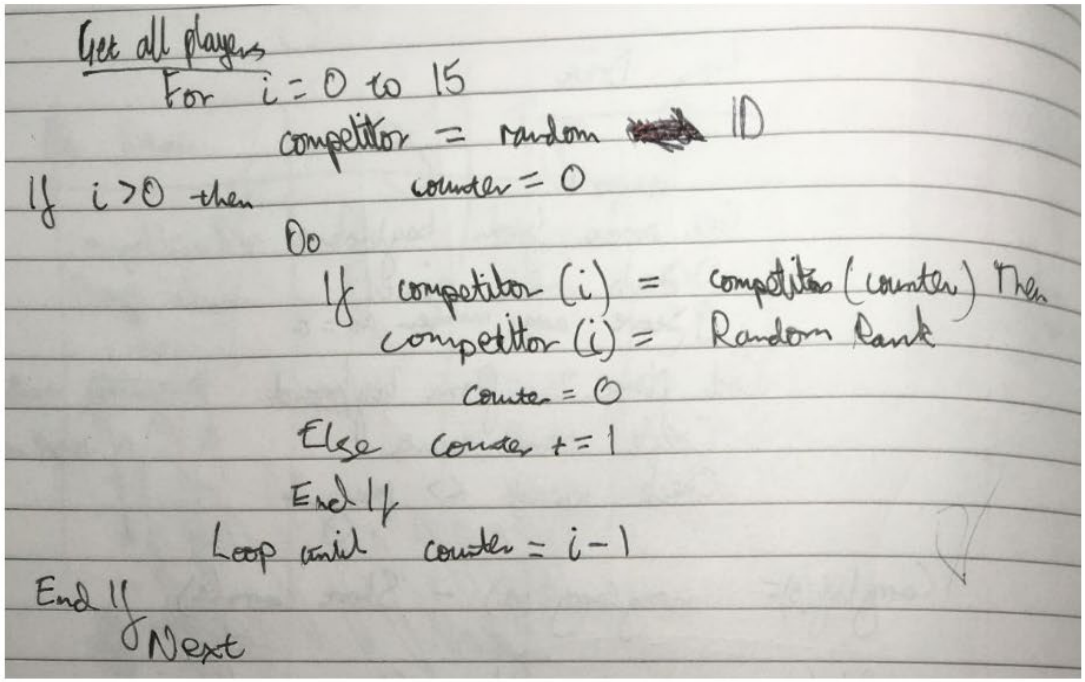
i = Uniconst - 1

Do Swap = false

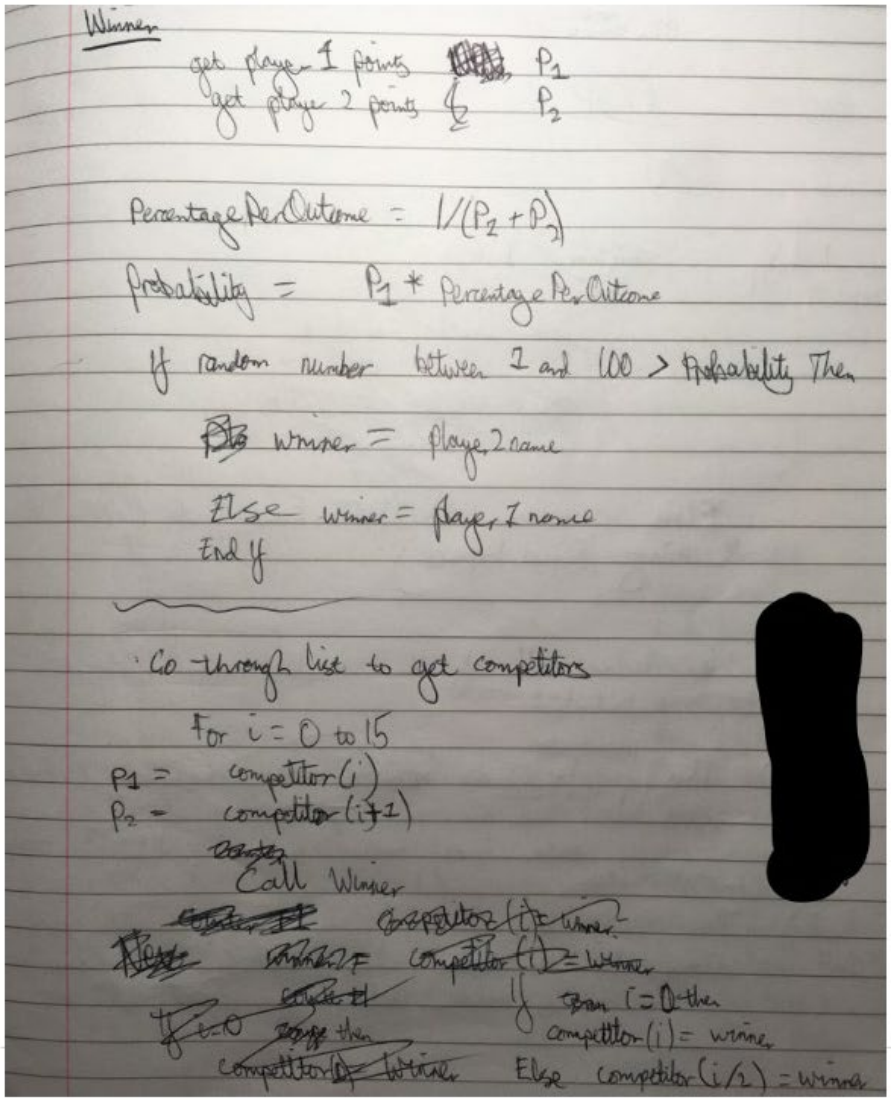
For j = 0 to i - 1

If player(j).averagepoints > player(j+1).averagepoints then
more = player(j)
player(j) = player(j+1)
player(j+1) = more
swap = true
End if
Next
i -= 1
Loop until i = 0 or swap = false

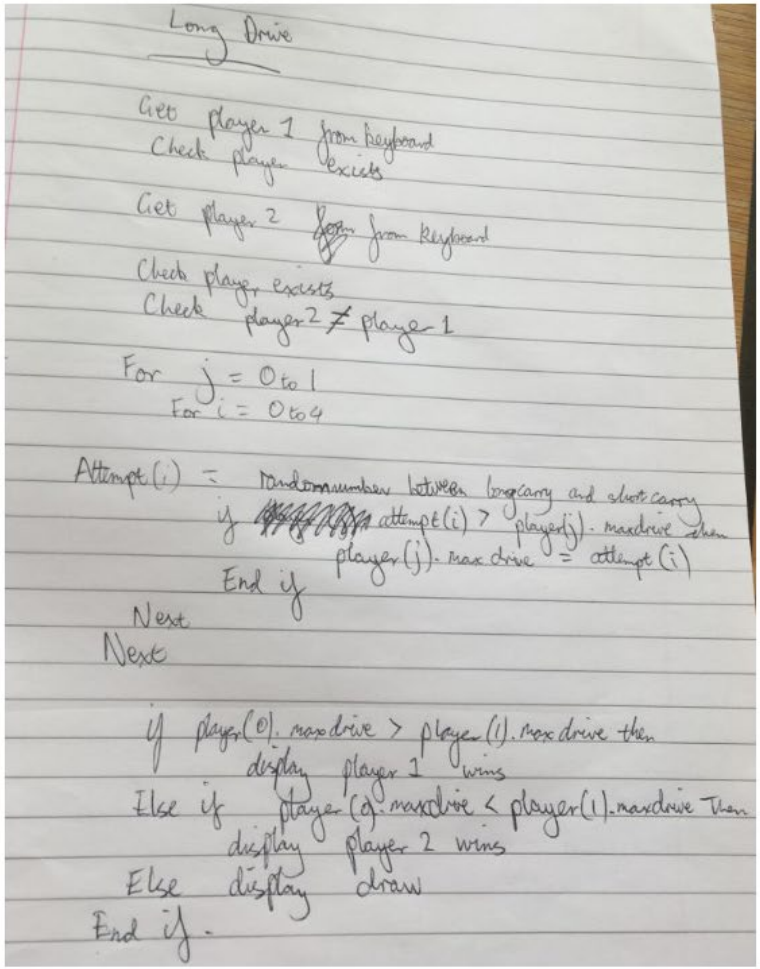
Matchplay Tournament



- *continued on next page*


Winner
get player 1 points P1
get player 2 points P2
PercentagePerOutcome = 1/(P1 + P2)
Probability = P1 * PercentagePerOutcome
If random number between 1 and 100 > Probability Then
Winner = player2name
Else winner = player1name
End If
Go through list to get competitors
For i = 0 to 15
P1 = competitor(i)
P2 = competitor(i+1)
Call Winner
If Ran [= 0 then
competitor(i) = winner
Else competitor(i/2) = winner

Long Drive competition



Long Drive

Get player 1 from keyboard
Check player exists

Get player 2 from keyboard

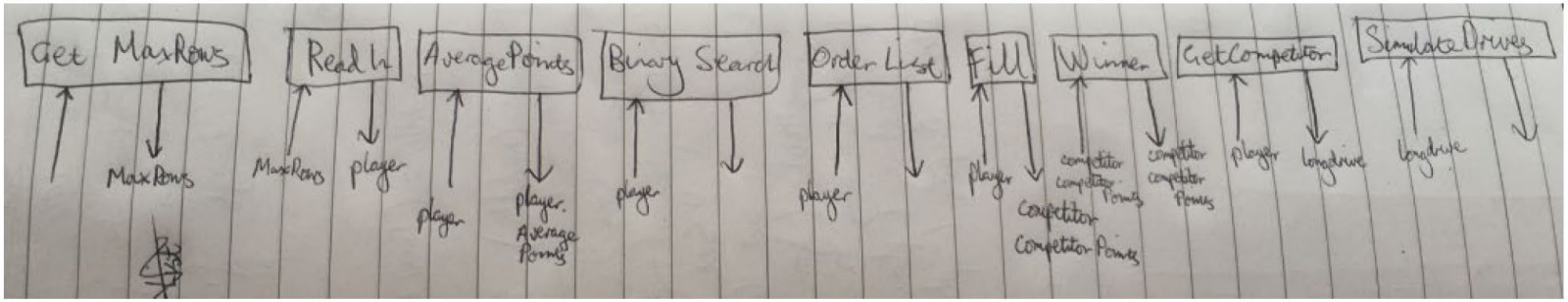
Check player exists
Check player2 ≠ player 1

For j = 0 to 1
For i = 0 to 4

Attempt(i) = random number between long carry and short carry
if attempt(i) > playerj). maxdrive then
player(j). max drive = attempt (i)
End if
Next
Next

if player(0). maxdrive > player (1). max drive then
display player 1 wins
Else if player (0. maxdrive < player(1). maxdrive Then
display player 2 wins
Else display draw
End if.

Dataflow


Get MaxRows
MaxRows
Read In
MaxRows
player
AverageScore
player
player.
Average
Points
Binary Search
player
Order List
player
Fill
player
Competitor
Competitor Points
Winner
competitor
competitor Points
competitor
competitor
Points
GetCompetitor
player
longdrive
SimulateDrives
longdrive

Implementation

Code

```
Imports System.IO
Public Class Form1

    'This outlines the fields in the array of records for the golfer's data
    Structure golfer
        Dim PlayerID As Integer
        Dim Name As String
        Dim EventsPlayed As Integer
        Dim Points As Integer
        Dim NumberOfWins As Integer
        Dim LongestCarry As Decimal
        Dim ShortestCarry As Decimal
        Dim Age As Integer
        Dim AveragePoints As Decimal
    End Structure


    'This finds the number of rows in the database
    Public Sub getmaxrows(ByRef MaxRows As Integer)
        Dim con As New OleDb.OleDbConnection


        'The database opens
        con.ConnectionString = "PROVIDER=Microsoft.ACE.OLEDB.12.0;Data Source ='\\     \Users\     \Desktop\Adv Higher
Assignment\GolfersDatabase.accdb'"
        con.Open()

        Dim ds As New DataSet
        Dim da As OleDb.OleDbDataAdapter
        Dim sql As String
```

```
        'The data is selected using an SQL query
        sql = "SELECT * FROM PGATOUR_data"

        da = New OleDb.OleDbDataAdapter(sql, con)
        da.Fill(ds, "Players")


        'This counts the number of rows in the database and stores them in the variable "MaxRows"
        MaxRows = ds.Tables("Players").Rows.Count


        'The database closes
        con.Close()
    End Sub


    'this reads in all of the data and stores it in the array of records called player
    Public Sub readIn(ByRef player() As golfer, ByVal MaxRows As Integer)

        Dim con As New OleDb.OleDbConnection


        'The database opens
        con.ConnectionString = "PROVIDER=Microsoft.ACE.OLEDB.12.0;Data Source ='\\    \Users\    \Desktop\Adv Higher
Assignment\GolfersDatabase.accdb'"
        con.Open()

        Dim ds As New DataSet
        Dim da As OleDb.OleDbDataAdapter
        Dim sql As String


        'The data is selected using an SQL query
        sql = "SELECT * FROM PGATOUR_data"

        da = New OleDb.OleDbDataAdapter(sql, con)
        da.Fill(ds, "Players")
```

```
    'The data is read into the array of records called Player
    For i = 0 To MaxRows - 1
        player(i).PlayerID = ds.Tables("Players").Rows(i).Item(0)
        player(i).Name = ds.Tables("Players").Rows(i).Item(1)
        player(i).EventsPlayed = ds.Tables("Players").Rows(i).Item(2)
        player(i).Points = ds.Tables("Players").Rows(i).Item(3)
        player(i).NumberOfWins = ds.Tables("Players").Rows(i).Item(4)
        player(i).LongestCarry = ds.Tables("Players").Rows(i).Item(5)
        player(i).ShortestCarry = ds.Tables("Players").Rows(i).Item(6)
        player(i).Age = ds.Tables("Players").Rows(i).Item(7)
    Next


    'The database closes
    con.Close()

End Sub



Function AveragePoints(ByVal player() As golfer, ByVal MaxRows As Integer)

    For i = 0 To MaxRows - 1
        'The player's average points is calculated and is stored in the array of records
        player(i).AveragePoints = player(i).Points / player(i).EventsPlayed
    Next
End Function



'This finds information about a player that a user has searched for
Public Sub BinarySearch(ByVal player() As golfer, ByVal MaxRows As Integer)

    Dim found As Boolean
    Dim target As Integer
```

```
        Dim low As Integer = 0
        Dim high As Integer = MaxRows - 1
        Dim middle As Integer
        Dim i As Integer = -1

        found = False


       'The user is asked for the player's ID that they want to search for
        target = InputBox("Enter player's ID to search for.")

        Do
            'The middle value is calculated
            middle = (low + high) \ 2

            'If the middle value is the one asked for by the user then found is true. The value of i is stored
            If player(middle).PlayerID = target Then
                found = True
                i = middle
                'If the value is too high then the highest value becomes the value below the highest value
            ElseIf player(middle).PlayerID > target Then
                high = middle - 1
                'If the middle value is too low then the lowest value is the value above the middle
            Else
                low = middle + 1
            End If


            'The loop is continued until the value is found or low becomes biggesr than high which means the value isn't on
the list else the details of the player are displayed
        Loop Until (found = True) Or (low > high)


        'If the value of i is less than 0 then it is output that the value isn't on the list
        If i < 0 Then
            MsgBox("The player you have entered is not on the list.")
        Else
```

```
            MsgBox(player(middle).Name & " is " & player(middle).Age & " years old. He had " & player(middle).NumberOfWins &
" wins in 2017 and played in " & player(middle).EventsPlayed & " events.")
        End If

    End Sub



    'This orders the players in order of their average points per tournament
    Public Sub OrderList(ByVal player() As golfer, ByVal MaxRows As Integer)

        Dim movePoints As Integer
        Dim moveName
        Dim swap As Boolean
        Dim i As Integer

        i = MaxRows - 1
        Do
            swap = False
            For j = 0 To i - 1
                'if the value of average points is biger than the value after it then they swap places
                If player(j).AveragePoints > player(j + 1).AveragePoints Then
                    movePoints = player(j).AveragePoints
                    moveName = player(j).Name
                    player(j).AveragePoints = player(j + 1).AveragePoints
                    player(j).Name = player(j + 1).Name
                    player(j + 1).AveragePoints = movePoints
                    player(j + 1).Name = moveName
                    swap = True
                End If
            Next
            i -= 1

            'this loop runs until there are no changes in the list
        Loop Until i = 0 Or swap = False

        For counter = 0 To MaxRows - 1
```

```
            ListBox2.Items.Add(player(counter).Name & vbTab & player(counter).AveragePoints)
        Next
    End Sub



    'Once the Sort and Search button is clicked these subprograms are run
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click

        Dim MaxRows As Integer
        'This assigns a value to MaxRows
        Call getmaxrows(MaxRows)

        'This declares the array of records
        Dim player(MaxRows - 1) As golfer
        'The values are read in
        Call readIn(player, MaxRows)
        'Average points are calculated
        Call AveragePoints(player, MaxRows)

        'The binary search carried out
        Call BinarySearch(player, MaxRows)
        'The order list is called
        Call OrderList(player, MaxRows)

    End Sub



    'This selects the 16 players that are to compete in the tournament
    Public Sub Fill(ByVal player() As golfer, ByRef competitor() As String, ByRef competitorPoints() As Decimal)
        Dim counter As Integer
        Dim number As Integer

        For j = 0 To 15
            'A random number is generated and a corresponding player ID is used to store the player's name and average
points
```

```
            Randomize()
            number = Int(194 * Rnd())
            competitor(j) = player(number).Name
            competitorPoints(j) = player(number).AveragePoints
            If j > 0 Then
                counter = 0
                Do
                    'This checks if the competitor has already been selected and generates another random player if he has
been
                    If competitor(j) = competitor(counter) Then
                        Randomize()
                        number = Int(194 * Rnd())
                        competitor(j) = player(number).Name
                        competitorPoints(j) = player(number).AveragePoints
                        counter = 0
                    Else : counter += 1
                    End If

                Loop Until counter = j
            End If
        Next
    End Sub


    'This finds the winner of the match between two players
    Public Sub Simulate(ByVal competitor() As String, ByVal competitorPoints() As Decimal, ByVal i As Integer, ByRef Winner
As String, ByRef WinnerPoints As Decimal)

        Dim P1 As Decimal
        Dim Name1 As String
        Dim P2 As Decimal
        Dim Name2 As String
        Dim PercentagePerOutcome As Decimal
        Dim ProbabilityOf1 As Decimal


        'The two players' names and average points are store
        P1 = competitorPoints(i)
```

```
        Name1 = competitor(i)
        P2 = competitorPoints(i + 1)
        Name2 = competitor(i + 1)

        'This calculates the percentage of and outcome
        PercentagePerOutcome = 1 / (P1 + P2)

        'This calculates the percentage chance of player 1 winning
        ProbabilityOf1 = (P1 * PercentagePerOutcome) * 100

        'A random number is generated and if it is greater than the percentage chance of player 1 then player 2 wins.
        If CInt(Math.Ceiling(Rnd() * 100)) > ProbabilityOf1 Then
            Winner = Name2
            WinnerPoints = P2
        Else : Winner = Name1
            WinnerPoints = P1
        End If
    End Sub



    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click

        Dim MaxRows As Integer
        'This assigns a value to MaxRows
        Call getmaxrows(MaxRows)
        'This declares the array of records
        Dim player(MaxRows - 1) As golfer
        'The values are read in
        Call readIn(player, MaxRows)
        'Average points are calculated
        Call AveragePoints(player, MaxRows)

        Dim competitor(15) As String
        Dim competitorPoints(15) As Decimal
        Dim Winner As String
        Dim WinnerPoints As Decimal
```

```
        Dim i As Integer = 0
        Dim round As Integer = 0

        Dim TextBox As TextBox
        Dim TextBoxName As String


        For round = 0 To 4
            'If it is the first round then the competitors names are generated
            If round = 0 Then

                'This generates the competitors
                Call Fill(player, competitor, competitorPoints)

                'This assigns their names to their corresponding textboxes
                For counter = 1 To 16
                    TextBoxName = "TextBox" & counter.ToString
                    TextBox = Me.Controls.Item(TextBoxName)
                    TextBox.Text = competitor(counter - 1)
                Next

            ElseIf round = 1 Then

                'The winner of each match is determined
                For i = 0 To 15
                    Call Simulate(competitor, competitorPoints, i, Winner, WinnerPoints)
                    If i = 0 Then
                        competitor(i) = Winner
                        competitorPoints(i) = WinnerPoints
                        'the i is divided by two to ensure they are placed at the next available location in the next round
on the competitors list. This value will overwrite the previous value that was stored there for the last round
                    Else : competitor(i \ 2) = Winner
                        competitorPoints(i \ 2) = WinnerPoints
                    End If

                    i += 1
                Next
```

```
                'This assigns the winner's names to their corresponding textboxes
                For counter = 17 To 24
                    TextBoxName = "TextBox" & counter.ToString
                    TextBox = Me.Controls.Item(TextBoxName)
                    TextBox.Text = competitor(counter - 17)
                Next

            ElseIf round = 2 Then

                i = 0

                'The winner of each match is determined
                For i = 0 To 7
                    Call Simulate(competitor, competitorPoints, i, Winner, WinnerPoints)
                    If i = 0 Then
                        competitor(i) = Winner
                        competitorPoints(i) = WinnerPoints
                        'the i is divided by two to ensure they are placed at the next available location in the next round
on the competitors list. This value will overwrite the previous value that was stored there for the last round
                    Else : competitor(i / 2) = Winner
                        competitorPoints(i / 2) = WinnerPoints
                    End If
                    i += 1
                Next

                'This assigns the winner's names to their corresponding textboxes
                For counter = 25 To 28
                    TextBoxName = "TextBox" & counter.ToString
                    TextBox = Me.Controls.Item(TextBoxName)
                    TextBox.Text = competitor(counter - 25)
                Next

            ElseIf round = 3 Then

                i = 0

                'The winner of each match is determined
```

```
                For i = 0 To 3
                    Call Simulate(competitor, competitorPoints, i, Winner, WinnerPoints)
                    If i = 0 Then
                        competitor(i) = Winner
                        competitorPoints(i) = WinnerPoints
                        'the i is divided by two to ensure they are placed at the next available location in the next round
on the competitors list. This value will overwrite the previous value that was stored there for the last round
                    Else : competitor(i / 2) = Winner
                        competitorPoints(i / 2) = WinnerPoints
                    End If
                    i += 1
                Next

                'This assigns the winner's names to their corresponding textboxes
                For counter = 29 To 30
                    TextBoxName = "TextBox" & counter.ToString
                    TextBox = Me.Controls.Item(TextBoxName)
                    TextBox.Text = competitor(counter - 29)
                Next

            Else
                i = 0
                'This simulates the final match
                Call Simulate(competitor, competitorPoints, i, Winner, WinnerPoints)
                'This assigns the competition winner's name to the textbox
                TextBox31.Text = Winner
                'This displays a message which tells the winner who has won the tournament
                MsgBox("The winner is " & Winner & ".")
            End If
        Next
    End Sub



    'This outlines the fields that will be used in the array of records for the long drive competition
    Structure LD
        Dim playerID As Integer
        Dim PlayerName As String
```

```
        Dim LongCarry As Decimal
        Dim ShortCarry As Decimal
        Dim maxDrive As Integer
    End Structure



    'This asks the user to enter two competitors that are not the same. This then stores the players' data in the long drive
array of records
    Public Sub GetCompetitors(ByRef LongDrive() As LD, ByVal player() As golfer)

        'This asks the user for the first player by selecting a player ID
        LongDrive(0).playerID = InputBox("Enter the first competitor's ID.")

        'This makes sure that the ID is valid
        Do
            If LongDrive(0).playerID > 195 Or LongDrive(0).playerID < 1 Then
                LongDrive(0).playerID = InputBox("Please enter a valid player ID.")
            End If
        Loop Until LongDrive(0).playerID > 0 And LongDrive(0).playerID <= 195

        'This reads in the data into the array of records
        LongDrive(0).PlayerName = player(LongDrive(0).playerID - 1).name
        LongDrive(0).LongCarry = player(LongDrive(0).playerID - 1).LongestCarry
        LongDrive(0).ShortCarry = player(LongDrive(0).playerID - 1).ShortestCarry

        'This asks the user for the second player by selecting a player ID
        LongDrive(1).playerID = InputBox("Enter the second competitor's ID.")

        Do
            'This makes sure that the ID is valid
            If LongDrive(1).playerID > 195 Or LongDrive(1).playerID < 1 Then
                LongDrive(1).playerID = InputBox("Please enter a valid player ID.")
                'This makes sure that the same player has not been entered as player 1
            ElseIf LongDrive(1).playerID = LongDrive(0).playerID Then
                LongDrive(1).playerID = InputBox("Please enter a different player ID to the first payer ID.")
            End If
```

```
        Loop Until LongDrive(1).playerID > 0 And LongDrive(1).playerID <= 195 And LongDrive(1).playerID <>
LongDrive(0).playerID

        'This reads in the data into the array of records
        LongDrive(1).PlayerName = player(LongDrive(1).playerID - 1).name
        LongDrive(1).LongCarry = player(LongDrive(1).playerID - 1).LongestCarry
        LongDrive(1).ShortCarry = player(LongDrive(1).playerID - 1).ShortestCarry

        'This displays the two player's names in a message box
        MsgBox(LongDrive(0).PlayerName & " vs " & LongDrive(1).PlayerName)

    End Sub



    'This simultates the long drive competition and displays the result
    Public Sub SimulateDrives(ByVal LongDrive() As LD)

        'This creates an array that will be used to store the value of the competitor's attempt
        Dim attempt(4) As Integer

        'This loop is run twice. Once for each competitor
        For j = 0 To 1
            'This loop runs 5 times. Once for each attempt
            For i = 0 To 4
                Randomize()
                'This generates a random number between the longest and shortest carries
                attempt(i) = Int((LongDrive(j).LongCarry - LongDrive(j).ShortCarry + 1) * Rnd() + LongDrive(j).ShortCarry)
                'This overwrites the maximum drive if the new value is higher
                If attempt(i) > LongDrive(j).maxDrive Then
                    LongDrive(j).maxDrive = attempt(i)
                End If
            Next
            'This adds the player's name and longest drive to the list box
            ListBox1.Items.Add(LongDrive(j).PlayerName & vbTab & LongDrive(j).maxDrive)
        Next
```

```
        'This displays in a message box the winner of the competition and their longest drive
        If LongDrive(0).maxDrive > LongDrive(1).maxDrive Then
            MsgBox(LongDrive(0).PlayerName & " wins with a drive length of " & LongDrive(0).maxDrive & ".")
        ElseIf LongDrive(0).maxDrive < LongDrive(1).maxDrive Then
            MsgBox(LongDrive(1).PlayerName & " wins with a drive length of " & LongDrive(1).maxDrive & ".")
        Else : MsgBox("Result is a draw!!!")
        End If

    End Sub



    'When the long drive button is clicked the the user is asked to input the player IDs of the competitors they want to
compete and then the tournament is simulated and results are output
    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click

        'This declares the long drive arrays of records
        Dim LongDrive(1) As LD

        Dim MaxRows As Integer
        'This assigns a value to MaxRows
        Call getmaxrows(MaxRows)
        'This declares the array of records
        Dim player(MaxRows - 1) As golfer
        'The values are read in
        Call readIn(player, MaxRows)

        'This runs the subprogram GetCompetitors
        Call GetCompetitors(LongDrive, player)
        'This runs the subprogram SimulateDrives
        Call SimulateDrives(LongDrive)
    End Sub


End Class
```

Screenshots

Database field names and types

PGATOUR_data

Field Name	Data Type
ID	AutoNumber
Player	Short Text
EVENTS_PLAYED	Number
POINTS	Number
NUMBER_OF_WINS	Number
LONGEST_CARRY_DISTANCE	Number
SHORTEST_CARRY_DISTANCE	Number
AGE	Number

The ID field

General | Lookup

Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Text Align	General

The Player Field

General | Lookup

Field Size	50
Format	@
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Allow Zero Length	Yes
Indexed	No
Unicode Compression	No
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

The Events_Played Field

General | Lookup

Field Size	Double
Format	General Number
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	No
Text Align	General

The POINTS field

General | Lookup

Field Size	Double
Format	General Number
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	No
Text Align	General

The Number_of_wins Field

General | Lookup

Field Size	Double
Format	General Number
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (Duplicates OK)
Text Align	General

The Longest_Carry_Distance field

General | Lookup

Field Size	Double
Format	General Number
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	No
Text Align	General

The Shortest_Carry_Distance field

General | Lookup

Field Size	Double
Format	General Number
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	No
Text Align	General

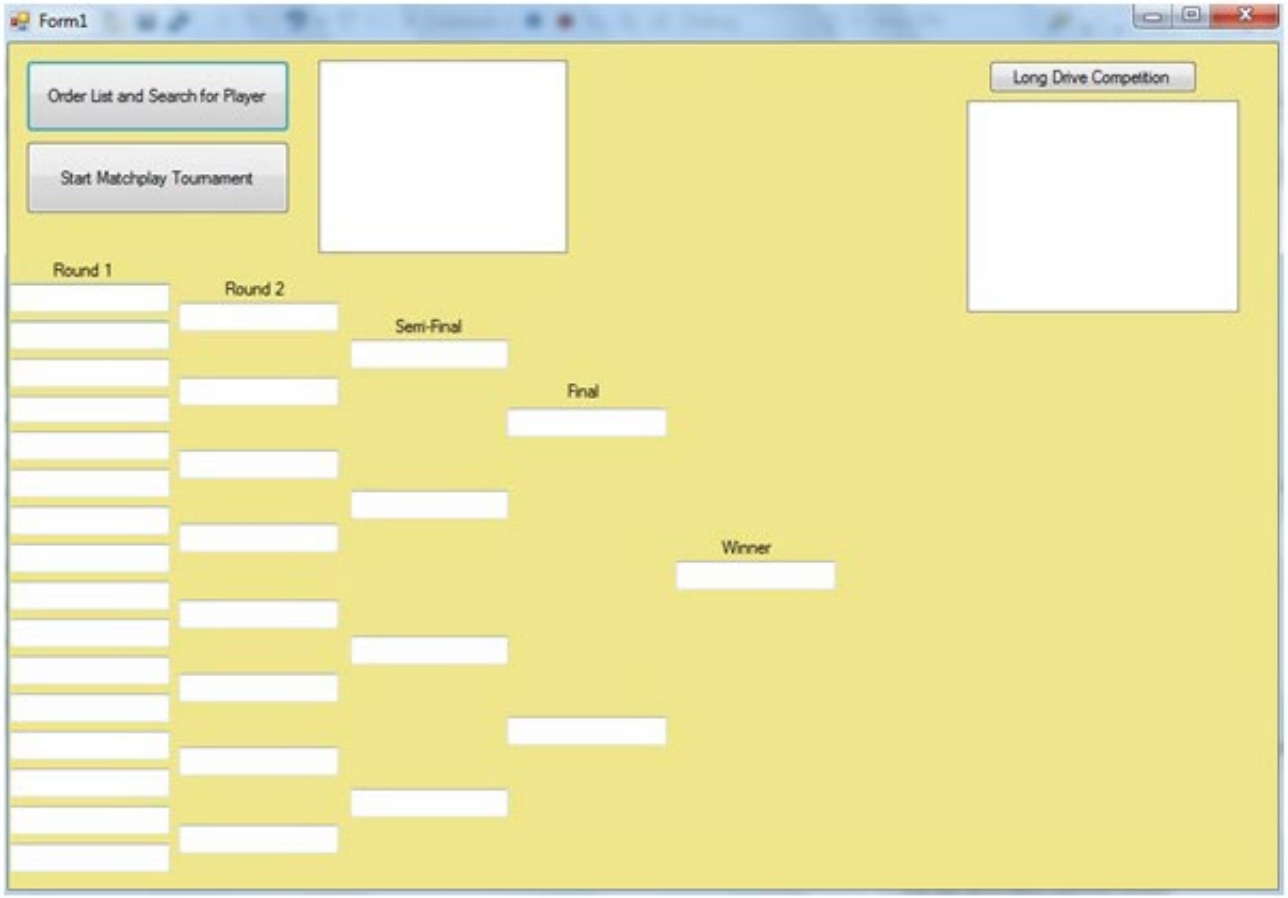
The AGE field

General | Lookup

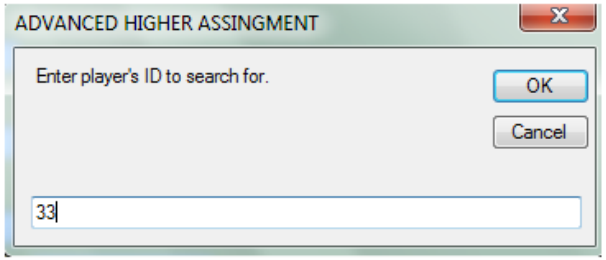
Field Size	Double
Format	General Number
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	No
Text Align	General

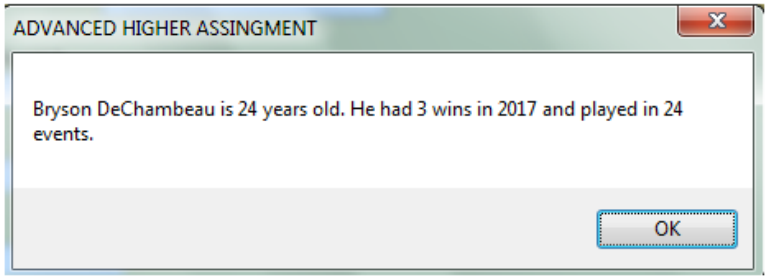
Program Screenshots

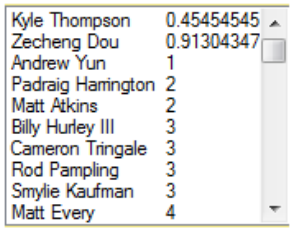
User Interface



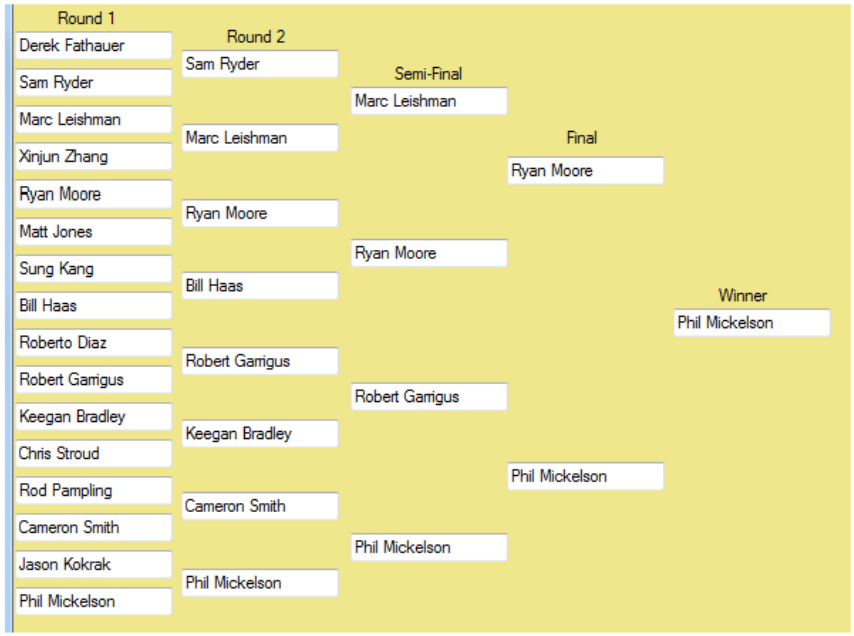
When "order list and search for players" button is clicked. A player ID is asked for and the corresponding player's details are displayed in a message box. The ordered list of average points in ascending order is displayed in the list box.

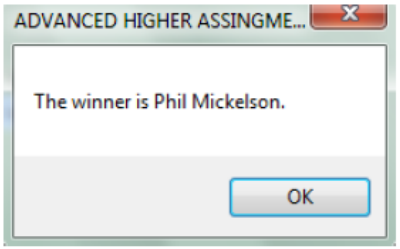






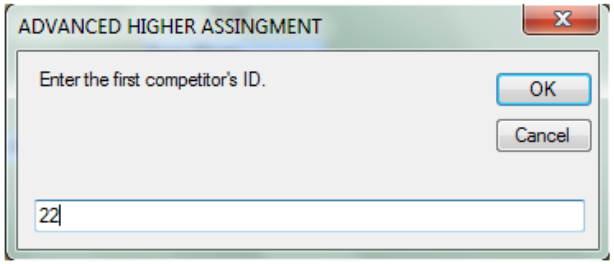
When "Start Matchplay Tournament" button is clicked, each match is simulated and displayed in the format of a knockout tournament. The winner's name is also displayed in a message box.

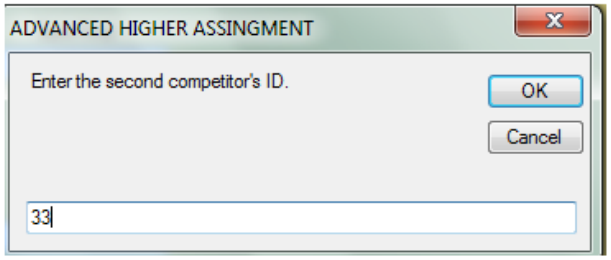




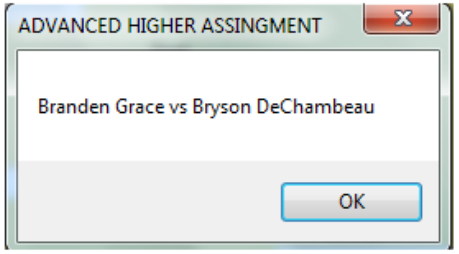
When "Long Drive Competition" button is clicked.

Player IDs are asked for.

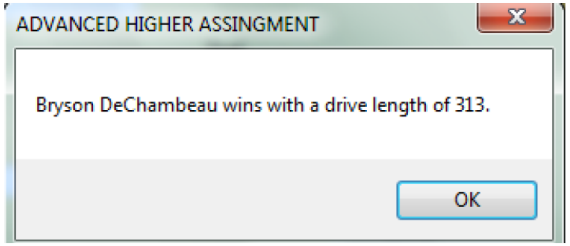


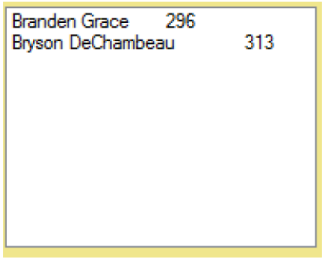


The corresponding players are displayed.



The result is displayed and the players' names and longest drives are added to the list box.





Ongoing testing

Problem 1

I am testing if the code to connect to a database is working. Stream writer code doesn't work for a database. This code would work for a csv file. An error is given as the file cannot be connected to this way. That means that there is no data to put in the array of records causing the error.

```
Dim strReader1 As StreamReader
 Dim lineCount = File.ReadAllLines("\\      \Users\     \Desktop\Adv Higher Assignment\GolfersDatabase.accdb").Length
      Dim player(lineCount - 1) As golfer
      Dim counter As Integer = 0
       Dim tempstring As String
      Dim temparray(7) As String
       strReader1 = File.OpenText("\\      \Users\      \Desktop\Adv Higher Assignment\GolfersDatabase.accdb")
       For counter = 0 To lineCount - 1
          tempstring = strReader1.ReadLine()
         temparray = tempstring.Split(",")
          player(counter).PlayerID = CInt(temparray(0))
        player(counter).Name = temparray(1)
        player(counter).EventsPlayed = CInt(temparray(2))
        player(counter).Points = CInt(temparray(3))
        player(counter).NumberOfWins = CInt(temparray(4))
        player(counter).LongestCarry = CDec(temparray(5))
        player(counter).ShortestCarry = CDec(temparray(6))
        player(counter).Age = CInt(temparray(7))
```

Next

strReader1.Close()

I used our class one note to find the answer of how to connect a program to a database.

Task

Download the Countries database and find the pathname of the file. Copy and paste this code, changing the path name to match.

```
        Dim con As New OleDb.OleDbConnection

        con.ConnectionString = "PROVIDER=Microsoft.ACE.OLEDB.12.0;Data Source
='\\Server\Users\.        \Desktop\countries.accdb'"
        con.Open()
        MessageBox.Show("Database is now open")


        Dim ds As New DataSet
        Dim da As OleDb.OleDbDataAdapter
        Dim sql As String

        sql = "SELECT * FROM Country"

        da = New OleDb.OleDbDataAdapter(sql, con)
        da.Fill(ds, "Countries")

        Dim MaxRows As Integer
        MaxRows = ds.Tables("Countries").Rows.Count

        For i = 0 To MaxRows - 1
            ListBox1.Items.Add(ds.Tables("Countries").Rows(i).Item(0))
        Next

        con.Close()
        MessageBox.Show("Database is now Closed")
```

I adapted the code for it to work in my program. The following code is the correct working code.

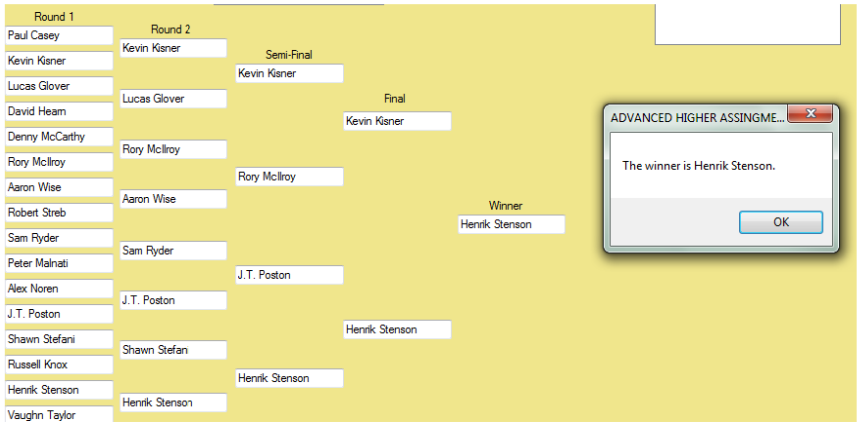
```
    'This outlines the fields in the array of records for the golfer's data
    Structure golfer
        Dim PlayerID As Integer
        Dim Name As String
        Dim EventsPlayed As Integer
        Dim Points As Integer
        Dim NumberOfWins As Integer
        Dim LongestCarry As Decimal
        Dim ShortestCarry As Decimal
        Dim Age As Integer
        Dim AveragePoints As Decimal
    End Structure


     'This finds the number of rows in the database
    Public Sub getmaxrows(ByRef MaxRows As Integer)
        Dim con As New OleDb.OleDbConnection

         'The database opens
        con.ConnectionString = "PROVIDER=Microsoft.ACE.OLEDB.12.0;Data Source ='\\     \Users\     \Desktop\Adv Higher
Assignment\GolfersDatabase.accdb'"
        con.Open()
         Dim ds As New DataSet
        Dim da As OleDb.OleDbDataAdapter
        Dim sql As String

         'The data is selected using an SQL query
        sql = "SELECT * FROM PGATOUR_data"
         da = New OleDb.OleDbDataAdapter(sql, con)
        da.Fill(ds, "Players")

         'This counts th3e number of rows in the database and stores them in the variable "MaxRows"
        MaxRows = ds.Tables("Players").Rows.Count

         'The database closes
        con.Close()
    End Sub
```

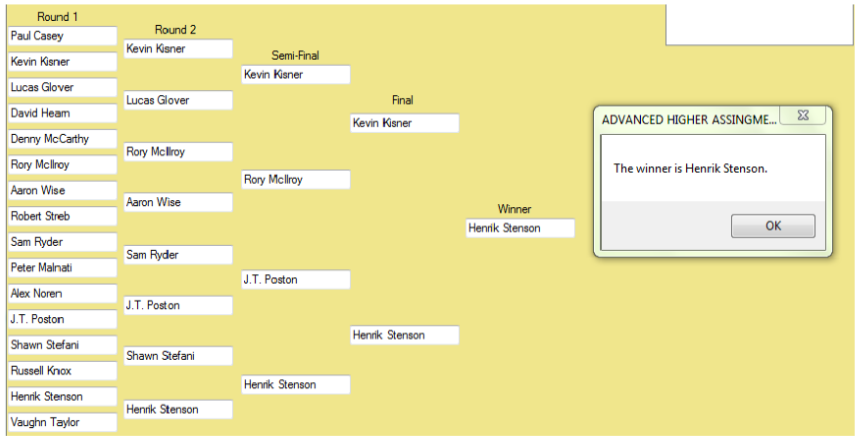
```
    'this reads in all of the data and store it in the array of records called player
    Public Sub readIn(ByRef player() As golfer, ByVal MaxRows As Integer)
         Dim con As New OleDb.OleDbConnection

         'The database opens
        con.ConnectionString = "PROVIDER=Microsoft.ACE.OLEDB.12.0;Data Source ='\\     \Users\     \Desktop\Adv Higher
Assignment\GolfersDatabase.accdb'"
        con.Open()
        Dim ds As New DataSet
        Dim da As OleDb.OleDbDataAdapter
        Dim sql As String

         'The data is selected using an SQL query
        sql = "SELECT * FROM PGATOUR_data"

        da = New OleDb.OleDbDataAdapter(sql, con)
        da.Fill(ds, "Players")

         'The data is read into the array of records called Player
        For i = 0 To MaxRows - 1
             player(i).PlayerID = ds.Tables("Players").Rows(i).Item(0)
            player(i).Name = ds.Tables("Players").Rows(i).Item(1)
            player(i).EventsPlayed = ds.Tables("Players").Rows(i).Item(2)
            player(i).Points = ds.Tables("Players").Rows(i).Item(3)
            player(i).NumberOfWins = ds.Tables("Players").Rows(i).Item(4)
            player(i).LongestCarry = ds.Tables("Players").Rows(i).Item(5)
            player(i).ShortestCarry = ds.Tables("Players").Rows(i).Item(6)
            player(i).Age = ds.Tables("Players").Rows(i).Item(7)
        Next

         'The database closes
        con.Close()
     End Sub


Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim MaxRows As Integer
        'This assigns a value to MaxRows
        Call getmaxrows(MaxRows)
          'This declares the array of records
        Dim player(MaxRows - 1) As golfer
```

```
        'The values are read in
        Call readIn(player, MaxRows)
End Sub
```

Problem 2

I am testing the results matchplay tournament. When I ran the code for the matchplay tournament I realised that I kept getting the same result every time. (see screenshots below).



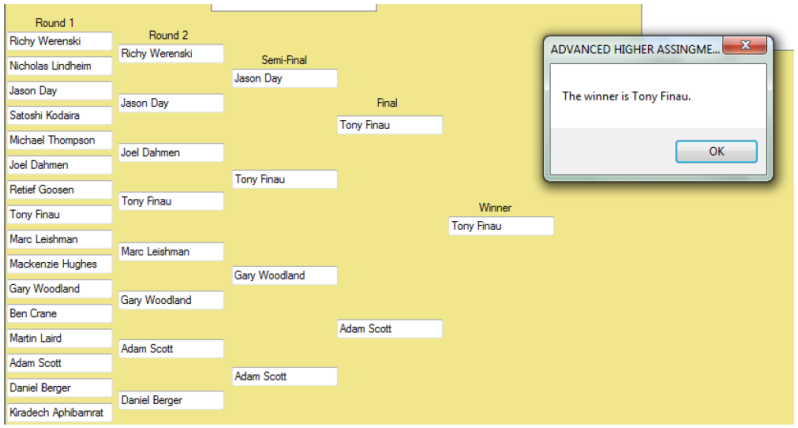


I researched online (http://ineasysteps.com/generating-random-number-visual-basic/) and found out that by using the RND() function the numbers were not really random. They were part of a sequence. This same sequence is used every time the program is started. I found out that by using the Randomize() function in the line before would mean that the sequence would change every time the code ran (see the highlighted code below).

```
For j = 0 To 15
            'A rondom number is generated and a corresponding player ID is used to store the player's name and average
points
            Randomize()
            number = Int(194 * Rnd())
            competitor(j) = player(number).Name
            competitorPoints(j) = player(number).AveragePoints
            If j > 0 Then
                counter = 0
                Do
                    'This checks if the competitor has already been selected and generates another random player if he has
been
                    If competitor(j) = competitor(counter) Then
                        Randomize()
                        number = Int(194 * Rnd())
                        competitor(j) = player(number).Name
                        competitorPoints(j) = player(number).AveragePoints
                        counter = 0
                    Else : counter += 1
                    End If
```

```
            Loop Until counter = j
        End If
        'ListBox1.Items.Add(competitor(j))
    Next
End Sub
```

The next output was different to the previous output.



External Course Research

External course http://www.vbforums.com/showthread.php?679024-RESOLVED-Is-it-possible-to-reference-a-text-control-using-a-variable

For the matchplay tournament I wanted to use textboxes in order show the progression of the tournament (e.g. The players in round 1 and then the players in round 2 etc.). In order to do this, after every round I had to assign the player's names to the corresponding textboxes. I could have done this by using the code:

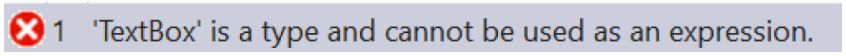
Textbox1.text = competitor(1)

Textbox2.text = competitor(2) etc.....

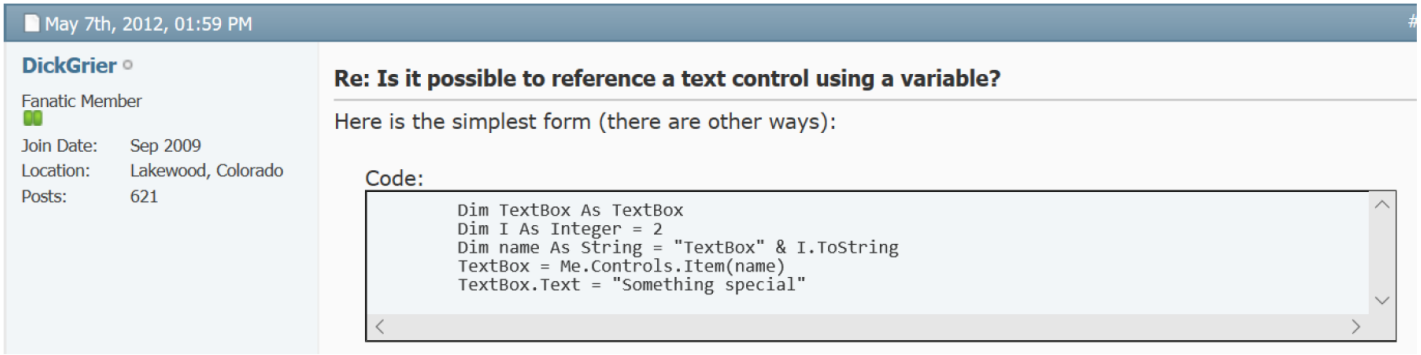
However this seemed very inefficient so I tried to use a fixed loop.

```
For i = 0 To 15
    TextBox(i).Text = competitor(i)
Next
```

But this gave me an error.



I researched how to assign textboxes values using a loop. This forum had the answer.
http://www.vbforums.com/showthread.php?679024-RESOLVED-Is-it-possible-to-reference-a-text-control-using-a-variable



May 7th, 2012, 01:59 PM

DickGrier

Fanatic Member

Join Date: Sep 2009
Location: Lakewood, Colorado
Posts: 621

Re: Is it possible to reference a text control using a variable?

Here is the simplest form (there are other ways):

Code:

```
Dim TextBox As TextBox
Dim I As Integer = 2
Dim name As String = "TextBox" & I.ToString
TextBox = Me.Controls.Item(name)
TextBox.Text = "Something special"
```

It showed me that by using the name of the textbox and storing it as a variable we can assign it new values. I adapted this code for my program.

```
For round = 0 To 4
            'If it is the first round then the competitors names are generated
            If round = 0 Then

                'This generates the competitors
                Call Fill(player, competitor, competitorPoints)

                'This assigns their names to their corresponding textboxes
                For counter = 1 To 16
                    TextBoxName = "TextBox" & counter.ToString
                    TextBox = Me.Controls.Item(TextBoxName)
                    TextBox.Text = competitor(counter - 1)
                Next

            ElseIf round = 1 Then

                'The winner of each match is determined
                For i = 0 To 15
```

```
                    Call Simulate(competitor, competitorPoints, i, Winner, WinnerPoints)
                    If i = 0 Then
                        competitor(i) = Winner
                        competitorPoints(i) = WinnerPoints
                        'the i is divided by two to ensure they are polaced at the next available location in the next round
on the competitors list. This value will overwrite the previous value that was stored there for the last round

                    Else : competitor(i \ 2) = Winner
                        competitorPoints(i \ 2) = WinnerPoints
                    End If

                    i += 1
                Next

                'This assigns the winner's names to their corresponding textboxes
                For counter = 17 To 24
                    TextBoxName = "TextBox" & counter.ToString
                    TextBox = Me.Controls.Item(TextBoxName)
                    TextBox.Text = competitor(counter - 17)
                Next

            ElseIf round = 2 Then
                i = 0

                'The winner of each match is determined
                For i = 0 To 7
                    Call Simulate(competitor, competitorPoints, i, Winner, WinnerPoints)
                    If i = 0 Then
                        competitor(i) = Winner
                        competitorPoints(i) = WinnerPoints

                        'the i is divided by two to ensure they are placed at the next available location in the next round
on the competitors list. This value will overwrite the previous value that was stored there for the last round
                    Else : competitor(i / 2) = Winner
                        competitorPoints(i / 2) = WinnerPoints
                    End If
                    i += 1
                Next

                'This assigns the winner's names to their corresponding textboxes
```

```
                For counter = 25 To 28
                    TextBoxName = "TextBox" & counter.ToString
                    TextBox = Me.Controls.Item(TextBoxName)
                    TextBox.Text = competitor(counter - 25)
                Next

            ElseIf round = 3 Then

                i = 0

                'The winner of each match is determined
                For i = 0 To 3
                    Call Simulate(competitor, competitorPoints, i, Winner, WinnerPoints)
                    If i = 0 Then
                        competitor(i) = Winner
                        competitorPoints(i) = WinnerPoints

                        'the i is divided by two to ensure they are polaced at the next available location in the next round
on the competitors list. This value will overwrite the previous value that was stored there for the last round
                    Else : competitor(i / 2) = Winner
                        competitorPoints(i / 2) = WinnerPoints
                    End If
                    i += 1
                Next

                'This assigns the winner's names to their corresponding textboxes
                For counter = 29 To 30
                    TextBoxName = "TextBox" & counter.ToString
                    TextBox = Me.Controls.Item(TextBoxName)
                    TextBox.Text = competitor(counter - 29)
                Next

            Else
                i = 0

                'This simulates the final match
                Call Simulate(competitor, competitorPoints, i, Winner, WinnerPoints)

                'This assigns the competition winner's name to the textbox
                TextBox31.Text = Winner
```

```
            'This displays a message which tells the winner who has won the tournament
            MsgBox("The winner is " & Winner & ".")
        End If
    Next
```

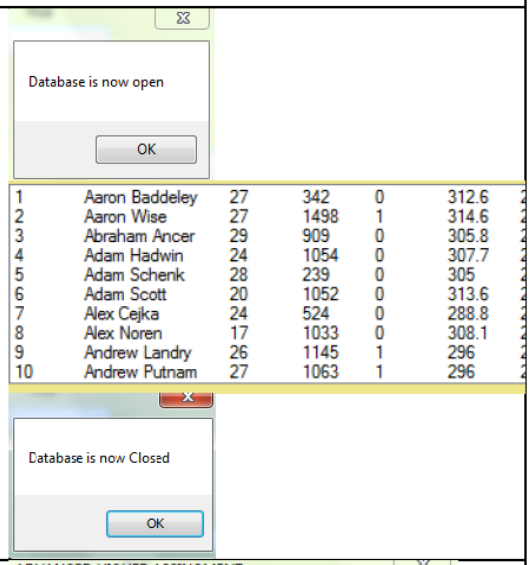
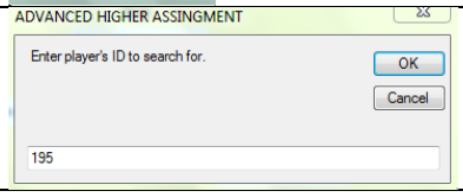
Testing

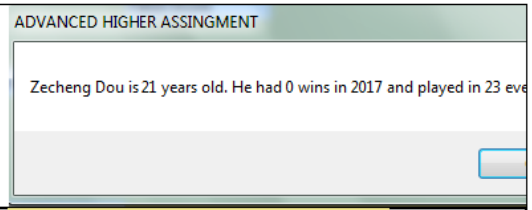
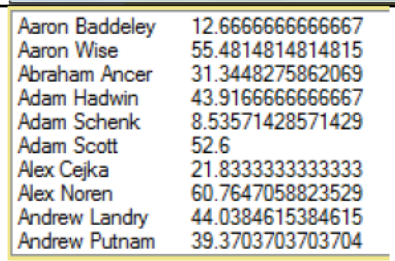
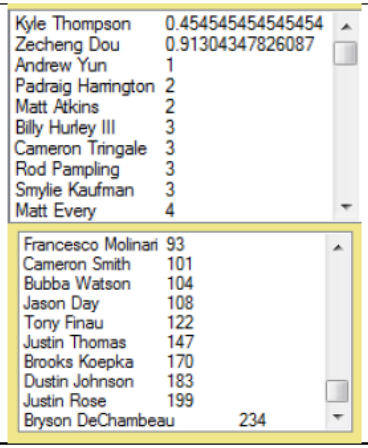
Final Test Plan

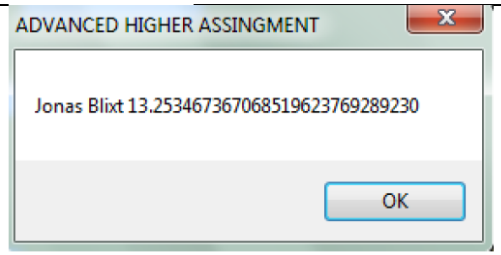
Test case ID	Test case objective	Test case description	Expected result
Module 1	Make sure data is read in from the database and stored in an array of records.	Create a listbox that shows all inputted data. Have message boxes to show when the database is open and closed	Database open is displayed. All fields are read in correctly. Aaron Baddeley is the first entry and Andrew Putnam should be entry 10. Database closed is displayed.
Module 2	Make sure the correct player name is found using the binary search algorithm.	Make sure message box displays the correct player and details.	The correct player is chosen. If 195 is entered then Zecheng Dou is displayed along with his age (21), number of wins (0) and events played (23).
Module 3	Make sure the players' average points are calculated correctly and are stored alongside the players' other details in the array of records.	Use a listbox to display the calculated average points. Calculate some of the players' average points by hand and compare.	Aaron Baddeley averages 12.6666667 points per tournament
Module 4	Make sure data is sorted in ascending order of average points using the bubble sort argorithm.	Go through listbox and ensure the players' names are in order of average points in ascending order.	All players are ordered by average score. Bryson DeChambeau is the last entry and Kyle Thompson is the first entry.
Module 5	Make sure 16 random names are generated. Ensure that there are no duplicate names.	Make sure that the player's matches are displayed in the correct textboxes. No duplicate names are selected.	One player is shown as the winner. No duplicate players.
Module 6	Ensure the tournament is simulated correctly using the average points to calculate the probability of the player winning the match and determine the winner for that match.	Use message boxes to show the probability of the player winning, the random number generated and the winner.	The player's name and probability will be displayed. The random number will then be displayed. If this is less than the probability then the player's name will be displayed. If not then the opponent's name will be displayed.

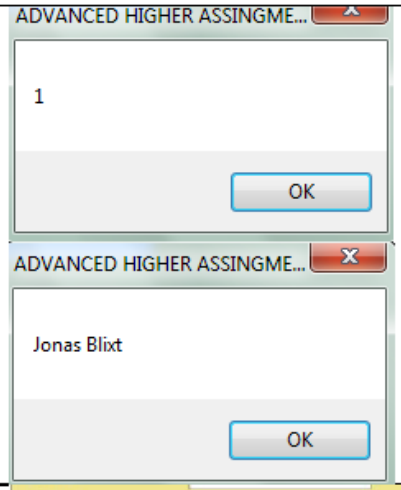
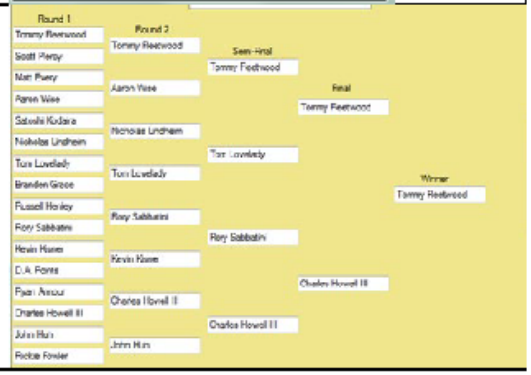
Module 7	Ensure the tournament players are displayed onscreen in the format of a knockout tournament in textboxes and the winner is displayed in a message box.	Check that the text boxes a completed and a message box with the correct name is displayed.	The players' routes are correct showing their competitors and where they have won and lost, just like in a standard knockout tournament. The winner's name displayed in a message box.
Module 8	Ensure that the two selected by the user for the long dive competition are valid.	Enter an ID that is out with the boundaries of the array. Enter the same ID twice.	The player should be asked to enter a valid ID. The user should be asked to enter a different ID to the first one entered.
Module 8	Ensure the two players selected by the user are displayed in a message box before the long drive competition and their names are added to the list box once the competition is simulated. Ensure the correct result is displayed in the message box.	Simulate a competition with player ID 1 and player ID 195 and record its outputs.	"Aaron Baddeley vs Zecheng Dou" should be displayed in a message box. Their names and their longest drive should be added to a list box. The correct result should be displayed in a message box. Either Aaron "Baddeley wins with a drive length of ..." or "Zecheng Dou wins with a drive length of ..." or "Result is a draw!!!".
Module 9	Check that the 5 distances are calculated for each player and the largest one is selected and compared with the other player's.	Add each attempt and the player's longest drive to a list box and ensure that the longest drive output is the largest attempt. Ensure that the message in the message box displays the correct result	The biggest attempt is the same as the longest drive. The correct player is shown as the winner or the result is a draw if both drives are the same length.

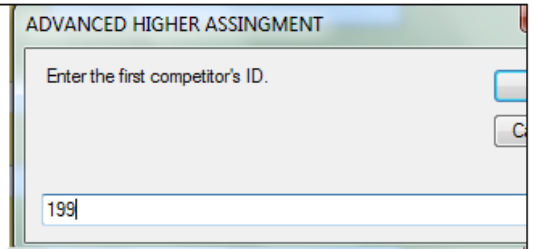
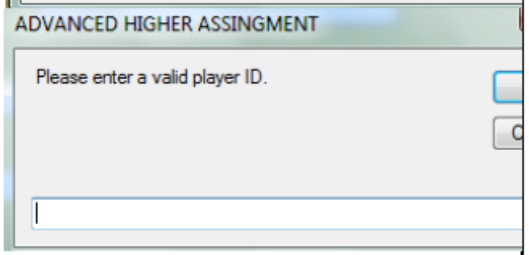
Requirements Testing

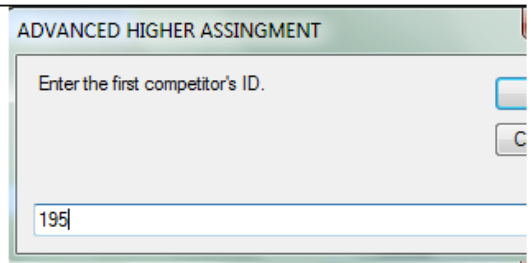
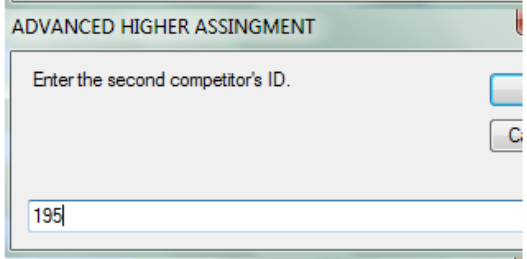
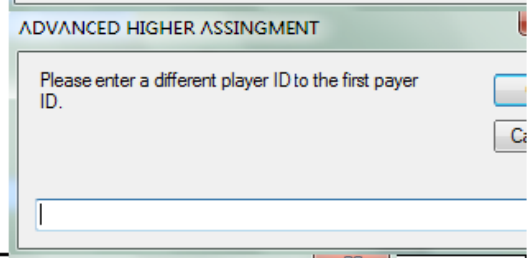
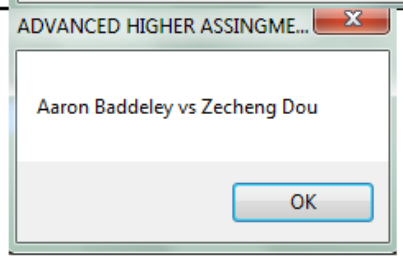
Test case ID	Test case objective	Test case description	Expected result	Actual Result
Module 1	Make sure data is read in from the database and stored in an array of records.	Create a listbox that shows all inputted data. Have message boxes to show when the database is open and closed	Database open is displayed. All fields are read in correctly. Aaron Baddeley is the first entry and Andrew Putnam should be entry 10. Database closed is displayed.	
Module 2	Make sure the correct player name is found using the binary search algorithm.	Make sure message box displays the correct player and details.	The correct player is chosen. If 195 is entered then Zecheng Dou is displayed along with his age (21), number of wins (0) and events played (23).	

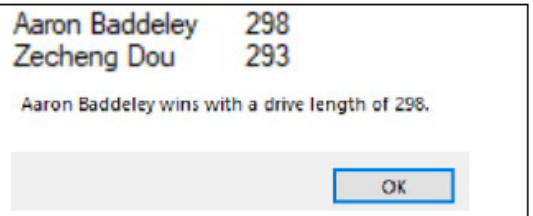
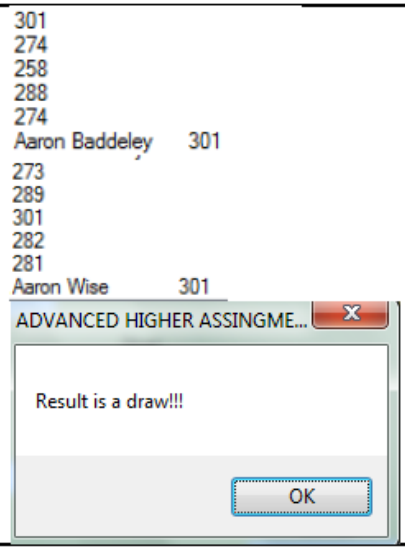
				
Module 3	Make sure the players' average points are calculated correctly and are stored alongside the players' other details in the array of records.	Use a listbox to display the calculated average points. Calculate some of the players' average points by hand and compare.	Aaron Baddeley averages 12.6666667 points per tournament	
Module 4	Make sure data is sorted in ascending order of average points using the bubble sort argorithm.	Go through listbox and ensure the players' names are in order of average points in ascending order.	All players are ordered by average score. Bryson DeChambeau is the last entry and Kyle Thompson is the first entry.	

Module 5	Make sure 16 random names are generated. Ensure that there are no duplicate names.	Make sure that the player's matches are displayed in the correct textboxes. No duplicate names are selected.	One player is shown as the winner. No duplicate players.	
Module 6	Ensure the tournament is simulated correctly using the average points to calculate the probability of the player winning the match and determine the winner for that match.	Use message boxes to show the probability of the player winning, the random number generated and the winner.	The player's name and probability will be displayed. The random number will then be displayed. If this is less than the probability then the player's name will be displayed. If not then the opponent's name will be displayed.	

				
Module 7	Ensure the tournament players are displayed onscreen in the format of a knockout tournament in textboxes and the winner is displayed in a message box.	Check that the text boxes a completed and a message box with the correct name is displayed.	The players' routes are correct showing their competitors and where they have won and lost, just like in a standard knockout tournament. The winner's name displayed in a message box.	

Module 8	Ensure that the two selected by the user for the long dive competition are valid.	Enter an ID that is out with the boundaries of the array. Enter the same ID twice.	The player should be asked to enter a valid ID. The user should be asked to enter a different ID to the first one entered.	  ----------

				  
Module 8	Ensure the two players selected by the user are displayed in a message box before the long drive competition and their names are added to the list box once the competition is	Simulate a competition with player ID 1 and player ID 195 and record its outputs.	"Aaron Baddeley vs Zecheng Dou" should be displayed in a message box. Their names and their longest drive should be added to a list box. The correct result should be displayed in a message box. Either Aaron	

	simulated. Ensure the correct result is displayed in the message box.		"Baddeley wins with a drive length of ..." or "Zecheng Dou wins with a drive length of ..." or "Result is a draw!!!".	
Module 9	Check that the 5 distances are calculated for each player and the largest one is selected and compared with the other player's.	Add each attempt and the player's longest drive to a list box and ensure that the longest drive output is the largest attempt. Ensure that the message in the message box displays the correct result	The biggest attempt is the same as the longest drive. The correct player is shown as the winner or the result is a draw if both drives are the same length.	

Personas

- Craig who is a regular golfer who is familiar with the professional game and how it works. They want to have some fun by simulating competitions and finding information about players. He has lots of experience with computers as programming is one of his hobbies.

Craig found the program clearly signposted as the buttons were clearly labelled. He found it hard to find the player they wanted as he had to know the player's ID beforehand. Once Craig figured out the player's ID he thought the long drive competition was a great way to have fun. Craig enjoyed simulating tournament but wished that the tournament was done one round at a time with a pause in between as it would build up more suspense to find the winner.

- Betty who has never played golf before but is intrigued by the sport and wants to learn more about it and its players at the same time as having fun. Her job is a receptionist and so is used to using computers.

Betty found the program easy. She didn't know the rules of the matchplay tournament or the longest drive competition and wishes that these could be explained on the program. She is intrigued by the player's names being ordered by average points. She found it easy to find the best players on the tour for that season.

- Johnathan doesn't like golf. He has never played it and has no interest in it. He is an elderly man who has little experience of technology. He enjoys watching other sports such as football, rugby and tennis.

Despite Johnathan's lack of computer knowledge, he found the program simple to use as the buttons were well named as they were descriptive of their actions. He found the concept of simulating a tournament interesting and thought that the output was clear as all of the players' names were displayed. He thought that the long drive competition was fun as the results were random. He suggested that there should be some way of knowing what player IDs correspond to what player. He suggested that they were put in a list box.

Evaluation

Fitness for Purpose

My program is fit for purpose as it reads all of the golfer's names from an external Access database. This information is stored in an array of records. It stores the variables PlayerID, name, eventsPlayed, points, numberOfWins, shortCarry, longCarry and age. I coded a binary search algorithm that will search for the player ID that was input in by the user. The corresponding player's name, age, numberOfWins and EventsPlayed are displayed in a message box. Average Points are calculated and ordered in average points in ascending order using a bubble sort algorithm. The ordered list which consists of the players' names and average points is displayed in a list box.

The program generates 16 random numbers which are used to select 16 random players for the matchplay tournament. Validation is used to ensure that none of these players names appear more than once in the list of 16 players. Average points are used to calculate the probability of one player winning the match. A random number is generated which determines who wins the match. This is the Winners function and is called for every individual match in the tournament. The tournament players are displayed onscreen in the format of a knockout tournament in textboxes and the overall winner is displayed in a message box.

Two players' names are input by the user to go head to head in the long drive competition. The two players selected by the user are validated by using input validation for the long dive competition. This ensures the IDs entered are between 1 and 195 and are not the same as each other. The two competitors' names are displayed in a message box before the long drive competition is simulated. 5 attempts are generated by each player between their longest and shortest carries. The largest attempt for each player is selected and compared with the other. Their names and longest drives are added to a list box once simulated. The correct result is selected and displayed in a message box.

The results of my testing show that the program is in full working order. All inputs, processes and outputs have been checked and meet the functional requirements.

Maintainability

My program is very maintainable as I have used meaningful variable names throughout my project. This makes it clear what information the variables store and therefore make it easier to see what the code is doing. If anyone else was to try and edit my project they would be able to do it easily as meaningful variable names make my code very readable.

```
Structure golfer
    Dim PlayerID As Integer
    Dim Name As String
    Dim EventsPlayed As Integer
    Dim Points As Integer
    Dim NumberOfWins As Integer
    Dim LongestCarry As Decimal
    Dim ShortestCarry As Decimal
    Dim Age As Integer
    Dim AveragePoints As Decimal
End Structure
```

I also used internal commentary to explain what the code is doing at certain points in the program. This allows people to clearly see what the code is used for and why it is there. This is particularly useful for complicated areas of code where it is tricky to see the purpose of the code. This means that anyone will be able to edit my code.

```
Dim MaxRows As Integer
'This assigns a value to MaxRows
Call getmaxrows(MaxRows)

'This declares the array of records
Dim player(MaxRows - 1) As golfer
'The values are read in
Call readIn(player, MaxRows)
'Average points are calculated
Call AveragePoints(player, MaxRows)

'The binary search carried out
Call BinarySearch(player, MaxRows)
'The order list is called
Call OrderList(player, MaxRows)
```

I use indentation in my code in order to show what parts of my code correspond to each other. This means anyone can see what the code does and how it interacts with other lines of code.

```
For j = 0 To 15
    'A random number is generated and a corresponding player ID is used to store the player's name and average points
    Randomize()
    number = Int(194 * Rnd())
    competitor(j) = player(number).Name
    competitorPoints(j) = player(number).AveragePoints
    If j > 0 Then
        counter = 0
        Do
            'This checks if the competitor has already been selected and generates another random player if he has been
            If competitor(j) = competitor(counter) Then
                Randomize()
                number = Int(194 * Rnd())
                competitor(j) = player(number).Name
                competitorPoints(j) = player(number).AveragePoints
                counter = 0
            Else : counter += 1
            End If
        Loop Until counter = j
    End If
Next
```

My code was created in modules which each module doing a different task. This splits up the code into separate tasks. By doing this it is easy to see what part of the code does what task. Having modules means that the program can be worked on by many people at the same time. Each programmer can work on a separate module which then means that all of the modules can be pulled together to make on large program.

```
Function AveragePoints(ByVal player() As golfer, ByVal MaxRows As Integer)

    For i = 0 To MaxRows - 1
        'The player's average points is calculated and is stored in the array of records
        player(i).AveragePoints = player(i).Points / player(i).EventsPlayed

    Next
End Function
```

Robustness

My program is robust as a variety of different numerical inputs can be entered and the program will not crash. This is due to the input validation placed on the inputs of the user in the long drive competition button. However, if a string is entered into an input box that asks for an integer then the program will crash.