**Project : 40361**

For Learner Use:

|  |  |
| --- | --- |
| Surname of Learner | Naidu |
| Name of Learner | Kian |
| Learner ID | 0110105463084 |
| Student Number | KNPMB070 |
| Date of Project Given | 29/05/2020 |
| Location / Branch | Durban |

For Assessors Use:

|  |  |
| --- | --- |
| Name of Branch | Durban |
| Name of Facilitator |  |
| Name of Assessor |  |
| Date Marked / Assessed |  |
| Mark Allocation | 55 |
| Mark Obtained |  |
| Competency Status ( C / NYC) |  |

|  |  |
| --- | --- |
| Unit Standards |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Candidates Signature | C:\Users\KIAN\AppData\Local\Microsoft\Windows\INetCache\Content.Word\(KNPMB070)Kian Naidu.jpg | Date Submitted | 07/06/2020 |
| Assessors Signature |  | Date  Marked |  |

1. Explain the difference between the For loop and the While loop and give a coded example of each type of loop. [6]

|  |
| --- |
| In the for loop initialization, condition and adjustment statements are done in a single line which makes the loop easier to understand whereas in the while loop initialization needs to be done before the loop can begin.  Example of For loop:  for (int value = 1; value < 4; value += 1) // initialization, condition and adjustments in 1 line.  {  Console.WriteLine("This for loop will run {0} times", value);  }  Example of While loop:  int value = 1; // initialization done before loop  while (value < 5) //condition set at the start of the loop  {  Console.WriteLine("The value is currently {0}", value);  value += 1; //adjustment done in loop  } |

1. You are required to write a C# program using a Console Application with the following decision table. [10]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quantity < 10 | Y | N | N | N |
| Quantity < 50 | Y | Y | N | N |
| Quantity < 100 | Y | Y | Y | N |
| Discount | 5% | 10% | 15% | 20% |

The table below shows possible test inputs and expected outputs.

|  |  |
| --- | --- |
| Quantity Input | Discount Output |
| 9 | 5% |
| 57 | 15% |

Allow your application to accept numerical inputs and display a message showing the discount that will be received.

eg.**Display:**

Enter the quantity amount: 9

The discount you will receive is 5%

Write the code you would use to achieve this.

|  |
| --- |
| using System;  namespace ConsoleApp5  {  class Program  {  static void Main(string[] args)  {  Console.WriteLine("Enter the quantity amount: ");  int quantity = Convert.ToInt32(Console.ReadLine());  int discount;  discount = 0;  if (quantity < 10)  {  discount += 5;  }  if ((quantity < 50) && (quantity >= 10))  {  discount += 10;  }  if ((quantity < 100) && (quantity >= 50))  {  discount += 15;  }  Console.WriteLine("The discount you will recieve is {0}%", discount);  }  }  } |

1. Discuss two categories of testing and explain how they could be applied to your program in the previous question. [4]

|  |
| --- |
| Black-box testing treats the software as if it was a black box, only focusing on inputs and outputs. This can be done by noting the amount read by the code and the answer given then comparing if the output is logical compared to the input given. This is to root out GIGO problems.  White-box testing is when a testers’ knowledge of the system internals is used when testing to make sure that each function or method has proper test cases available.  This can be done by checking if all the if statements and conditions are correct and process the data correctly to give information |

1. Give the definition for the BubbleSort and QuickSort algorithm. [2]

|  |
| --- |
| BubbleSort: uses a series of comparison and swap operations to arrange the elements of a list in the correct order which can be either ascending or descending.  QuickSort: uses the partitioning and comparison operations to arrange the elements of a list in the correct order which can be either ascending or descending. |

1. You going to create a guessing game using a C# Console Application. [10]

Your application needs to generate a random integer between 1 and 10 (see the MSDN documentation for the how to create number with the Random class: http://msdn.microsoft.com/en-us/library/system.random.aspx).

Your application will then prompt user to enter a number - if the user has guessed the secret number a congratulatory message should be displayed otherwise the user needs to be told that they guessed too high or too low.

eg.**Display:**

Guess a number between 1 and 10: 3

You guessed too low.

Guess a number between 1 and 10: 9

You guessed too high

Guess a number between 1 and 10: 6

You guessed the secret number

Press any key to continue...

|  |
| --- |
| using System;  namespace RandomNumberGenerator  {  class Program  {  static void Main(string[] args)  {  Random random = new Random();  int randomnumber = random.Next(1, 11);  int usernumber = 0;  while (usernumber != randomnumber)  {  Console.WriteLine("Guess a number between 1 and 10" + randomnumber);  usernumber = Convert.ToInt32(Console.ReadLine());  if (usernumber < randomnumber)  {  Console.WriteLine("You guessed too low");  }  else if (usernumber > randomnumber)  {  Console.WriteLine("You guessed too high");  }  }  Console.WriteLine("You guessed the secret number");  }  }  } |

1. Consider the following code snippet: [4]

int n = 20;

int d = n++ 5;

What will be the value of d after this code snippet is executed? Give a detailed explanation in terms of unary operators.

1. 25
2. 26
3. 27
4. 28

|  |
| --- |
| ~~(Assuming int d = n++ + 5 since it’s the only logical answer based on choices given.)~~  A)  The value of n at the first line of the snippet is 20. The next line shows n++ which is a postfix unary operator, meaning the value of n will only be altered after it is used and currently is still 20 in the second line. So 20 added to 5 is 25 which is the value of d. |

1. This question relates to Arrays [10]
2. Create a console application called PeopleArray.
3. Create an array of strings of five people’s names.
4. Display the names of the people in the console window using TWO different iteration structures.
5. Display the message, “This person is the greatest”, next to the person that is third in the array list.

|  |
| --- |
| using System;  namespace Array  {  class Program  {  static void Main(string[] args)  {  string[] ArrPeople = new string[5] { "William", "Alice", "Kevin", "Christopher","Melia" };  for (int count = 0; count < 5; count += 1)  {  while (count == 2) //self-note: while loop slightly inefficient, but needed for criteria  {  Console.WriteLine(ArrPeople[count] + " This person is the greatest");  count += 1;  }  Console.WriteLine(ArrPeople[count]);  }  }  }  } |

1. Explain the purpose of the try…catch…finally block [3]

|  |
| --- |
| The purpose is to catch any exceptions that may be brought up in the software which could cause the program to terminate such as division by zero. The code that causes the exception should be put in the try block and then the code that handles the exceptions should be inside a catch block. The finally block will always execute regardless of whether there is an exception or not, this is often used to write clean up code. |

1. List and explain different types of testing levels. [4]

|  |
| --- |
| Unit testing, integration testing, system testing and acceptance testing.  Unit testing: verifies functionality of the unit of code.  Integration testing: assesses the interface between the software components is correct and functions correctly.  System testing: overall testing of the software system  Acceptance testing: tested by customers themselves which is further subdivided into alpha and beta testing. |

1. Differentiate between Black-box testing and white-box testing. [2]

|  |
| --- |
| Black-box testing just tests the inputs and outputs to ensure it is logical whereas white-box testing tests the functionality of the systems internals by those who have access to the source code. |

*\*\*END OF ASSESSMENT\*\**