**Excel Technologies Ltd.**

**Competency Assessment**

Position: Software Engineer

Total Marks: 30

*Ensure you are indicating accurately what question you are answering. Make sure your script is clean and easily readable. Do not forget to fill up the box below with your information. Attach the question paper with you script. During competency assessment, you are not allowed to use internet through any means.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | M | O | H | A | M | E | D | A | B | D | E | L | R | H | M | A | N |  |  |  |
| Cellphone | 0 | 0 | 9 | 6 | 6 | 5 | 4 | 8 | 7 | 9 | 4 | 7 | 1 | 9 |  |  |  |  |  |  |
| Email | M | O | H | A | S | A | E | E | D | K | S | A | @ | G | M | A | I | L | . | COM |

|  |  |  |
| --- | --- | --- |
|  |  | Marks |
| 1 | Consider the following register. Holy Family Red Cross Hospital is using this register to manage doctors’ list, their contact number, and the departments where the doctors are belongs to. With this register, the hospital is also managing doctor’s service points within the hospital.   1. Apply normalization rule to normalize this register up to 3rd normal form. 2. After normalization, draw Entity Relationship Diagram and show the degree of cardinality among entities using crow’s foot notation.  |  |  |  |  | | --- | --- | --- | --- | | **Doctor** | **Contact Number** | **Service Points** | **Department** | | Dr. Lissa Mwenda | +260766219936 | Antenatal Care, Family Planning, Postnatal Care | Gynecology | | Dr. Yvonne Sishuwa | +260766219937 | Family Planning, Postnatal Care | Pediatrics | | Dr. Machalo Mbale | +260766219938 | Antenatal Care | Radiology and Imaging |   1NF  TABLE 1 : Doctor , contact Number , Department  TABLE1 2 : Doctor, Service Point ,Department  2NF  TABLE 1: Doctor , contact Number , DeptId  TABLE 2 : Doctor, Service Point ,  TABLE 3 DeptId , Department  3NF  TABLE 1: Doctor , contact Number , DeptId  TABLE 2 : Doctor, DeptId ,Service Point  TABLE 3 DeptId , Department  1  m  Doctor  m  has  has  m  1  Service Point  1  has  Departments | 5 X 2 = 10 |
| 2 | Consider the following loop. Trace the value of “n” in every iteration of the loop.  int n = 30;  for (int i = 0; i <= 5; i++)  {  n += i;  }  print(n);  **value of “n” in every iteration of the loop**: 30 , 31 , 33 , 36 , 40 , 45 | 5 |
| 4 | Explain method overloading and method overriding with example. Write your code in C# programming language.  **Method Overloading** is the common way of implementing polymorphism. It is the ability to redefine a function in more than one form.  Method overloading can be done by changing :   1. By changing the Number of Parameters 2. By changing the Data types of the parameters 3. By changing the Order of the parameters   using System;  class MOVERLOAD  {  // adding two integer values.  public int Add(int a, int b)  {  int sum = a + b;  return sum;  }  // adding three integer values.  public int Add(int a, int b, int c)  {  int sum = a + b + c;  return sum;  }  // Main Method  public static void Main(String[] args)  {  // Creating Object  MOVERLOAD ob = new MOVERLOAD();  int sum1 = ob.Add(1, 2);  Console.WriteLine("sum of the two "  + "integer value : " + sum1);  int sum2 = ob.Add(1, 2, 3);  Console.WriteLine("sum of the three "  + "integer value : " + sum2);  }  }  **Method Overriding** is a technique that allows the invoking of functions from another class (base class) in the derived class. Creating a method in the derived class with the same signature as a method in the base class is called as method overriding.  using System;  class baseClass  {  // print() is 'virtual' here  public virtual void print()  {  Console.WriteLine("From Base class");  }  }  // class 'baseClass' inherit  // class 'derived'  class derived : baseClass  {  //'print()' is 'override' here  public override void print()  {  Console.WriteLine("From Derived class");  }  }  class MOVERIDE  {  // Main Method  public static void Main()  {  baseClass obj;  obj = new baseClass();  obj.print(); // output -> From Base class  // the same object 'obj' is now  // the object of class 'derived'  obj = new derived();  obj.print(); // output -> From Derived class  }  } | 5 |
| 5 | Translate the following UML Class Diagram into program code. Write your code in C# programming language. | 5 |
| using System;  using System.Collections.Generic;  using System.Text;  namespace ClinicianProject  {  public class Clinician  {  public string Name { get; set; }  public string HospitalName { get; set; }  public bool Login(string UserName, string Password)  {  // check UserName and Password here  return true;  }  private bool IsSessionExists(string UserName)  {  /// check session is exists  return true;  }  }  public class Doctor : Clinician  {  public string practiceNumber { get; set; }    public void createPrescription(int patientNumber)  {    }  }  public class Pharmacist : Clinician  {  public string PharmacistNumber { get; set; }  public void dispenceMedications(int prescriptionNumber)  {  }  }  } | | |
| 6 | Translate the UML Activity diagram into program code. Write your code either in C# programming language. | 5 |
|  | | |

public class GetMinNumber

{

// Main Method

public static void Main()

{

var Min = 0;

Console.Write("Please enter First Number, then press ENTER: ");

var userInput = Console.ReadLine();

var n1 = Convert.ToInt32(userInput);

Console.Write("Please enter Second Number, then press ENTER: ");

userInput = Console.ReadLine();

var n2 = Convert.ToInt32(userInput);

Console.Write("Please enter Third Number, then press ENTER: ");

userInput = Console.ReadLine();

var n3 = Convert.ToInt32(userInput); ;

if(n1 < n2)

{

Min = n1;

}

else

{

Min = n2;

}

if(n3 < Min)

{

Min = n3;

}

Console.Write("Min is : " + Min);

}

}

-END-