UNIVERSITY OF WESTMINSTER

INFORMATICS INSTITUTE OF TECHNOLOGY

Department of Computing

(B.Eng.) in Software Engineering

Module: 5COSC007C.1 Object Oriented Programming

First Name : **Premananth**

Surname : **Vishnugaran**

Student ID : **2018304**

Uow ID : **W1716294**

Module Leader : **Guhanathan P**

Submission Date : **12/01/2021**

Premananth Vishnugaran
2018304
Module: 5COSC007C.1

Question 1

City Entity Class

```
public class CityEntity {
String name;
   int restaurantCount;

   public CityEntity(String name, int restaurantCount) {
this.name = name;
      this.restaurantCount = restaurantCount;
   }

   public CityEntity() {
   }

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public int getRestaurantCount() {
return restaurantCount;
   }

   public void setRestaurantCount(int restaurantCount) {
this.restaurantCount = restaurantCount;
   }
   @Override
   public String toString() {
return "CityEntity{" +
            "name='" + name + '\'' +
            ", restaurantCount=" + restaurantCount +
            '}';
   }
```

```
}
```

Restaurant Entity Class

```
Public class RestaurantEntity {
   String name;
   String address;
String city;
   int menuCount;

   public RestaurantEntity() {
   }

   public RestaurantEntity(String name, String address, String city, int menuCount) {
this.name = name;        this.address = address;        this.city = city;
      this.menuCount = menuCount;
   }

   public String getName() {
return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public String getAddress() {
return address;
   }

   public void setAddress(String address) {
      this.address = address;
   }

   public String getCity() {
      return city;
   }

   public void setCity(String city) {
```

Premananth Vishnugaran
2018304
Module: 5COSC007C.1

```
        this.city = city;
    }

    public int getMenuCount() {
return menuCount;
    }

    public void setMenuCount(int menuCount) {
this.menuCount = menuCount;
    }

    @Override
    public String toString() {
return "RestaurantEntity{" +
"name='" + name + '\'' +
            ", address='" + address + '\'' +
            ", city='" + city + '\'' +
            ", menuCount=" + menuCount +
            '}';
    }
}
```

Premananth Vishnugaran
2018304

Customer Entity Class

```
import java.util.ArrayList;
import java.util.Arrays; import
java.util.Date;
import java.util.List;

public class CustomerEntity {

   String firstName;
   String surname;
   String dob;
   String employer;
   String[] restaurant = new String[3];

   public CustomerEntity() {
   }

   public CustomerEntity(String firstName, String surname, String dob, String employer, String[]
restaurant) {
      this.firstName = firstName;
this.surname = surname;
      this.dob = dob;
this.employer = employer;
      this.restaurant = restaurant;
   }

   public String getFirstName() {
return firstName;
   }

   public void setFirstName(String firstName) {
      this.firstName = firstName;
   }

   public String getSurname() {
return surname;
   }
```

```
    public void setSurname(String surname) {
this.surname = surname;
    }

    public String getDob() {
        return dob;
    }

    public void setDob(String dob) {
        this.dob = dob;
    }

    public    String    getEmployer()    {
return employer;
    }

    public void setEmployer(String employer) {
this.employer = employer;
    }

    public    String[]    getRestaurant()    {
return restaurant;
    }

    public void setRestaurant(String[] restaurant) {
this.restaurant = restaurant;
    }

    @Override
    public    String    toString()    {
return "CustomerEntity{" +
            "firstName='" + firstName + '\'' +
            ", surname='" + surname + '\'' +
            ", dob='" + dob + '\'' +
            ", employer='" + employer + '\'' +
            ", restaurant=" + Arrays.toString(restaurant) +
            '}';
    }
```

```
}
```

Question 2

Private : The access level of a private modifier is only within the class. It cannot be accessed from outside the class.

Public: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.

Protect : Protected: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package

Public class can access fields and methods declared.

Public – Main Method – Access Specifier

Question 3

Option A – Because always will contains characteristics of parent.

Option E – Because Object is instance of class

Premananth Vishnugaran
2018304
Module: 5COSC007C.1

Question 4

A Question

```
public class Book { private
String title; private String
author;
private int numOfPage;
}
```

B Question

```
 public Book(String title, String author, int numOfPae)
{  this.title = title;  this.author = author;
this.numOfPae = numOfPae;
 }
```

C Question

```
   public    int    getNumOfPage()    {
return numOfPage;
   }
```

D Question

```
public    boolean    isLonger(Book    book){
boolean flag;
      if (num_of_page > book.getNum_of_page()){
flag = true;
      }
else{
```

```
            flag = false;
        }
        return flag;
    }
```

Premananth Vishnugaran
2018304
Module: 5COSC007C.1

E Question

```
@Override
   public String toString() {       return "Book{" +"title='" + title
+ ',' +", author='" + author
```

F Question

```
public class TextBook extends Book{
String[] modules;
}
```

G Question

```
public TextBook(String title, String author, int num_of_pages, int num_of_modules) {
            super(title, author, num_of_pages);
this.modules=new String[num_of_modules];
             }
```

H Question

I Question

```
public static void
main(String args[]){
Book B1 = new TextBook("Ulyssses","James joyce",150,3);
Book B2 = new TextBook("Don Quixote","<iguel",90,2);
Book B3 = new TextBook("Moby Dick","Herman",70,3);

 System.out.println(B1);
 System.out.println(B2);
 System.out.println(B3);
}
```

Question 5

Extends Thread: your thread creates unique object and associate with it. The run() method is what is executed by the thread after you call start().

Implements Runnable: it shares the same object to multiple threads. The Runnable interface only has a single method run().

Extending Thread class will not give you an option to extend any other class. But if you implement Runnable interface you could extend other classes in your class.So depending on your design requirement

Example :

```
Extends Thread :    class ex extends Thread{
    public static void main(String args[]){                ex
t1=new ex();            t1.start();
                          } }
```

Question 6

Question 7

```
public static void main(String[] args) {
    String length;
    String[] letters
={"one","two","three","four","five","six","seven","eight","nine","ten","eleven","twelve","thirte
en","fourteen","fifteen","sixteen","seventeen","eighteen","nineteen","twenty","thirty","forty",
"fifty","sixty","seventy","eighty","ninety","hundred","thousand"};
    int total =0;
    int  oneToNine=0;          int
tenToNineteen=0;               int
twentyToNinetyNine = 0;        int
hundredToThousand=0;
    String and ="and";

    for(int i=0;i<9;i++){
       oneToNine = oneToNine + letters[i].length();
    }

    for(int i=9;i<19;i++){
       tenToNineteen = tenToNineteen +  letters[i].length();
    }

    for(int i=19;i<27;i++){
```

```
        twentyToNinetyNine = twentyToNinetyNine + letters[i].length();
    }

    total = (10*twentyToNinetyNine)+ 8*oneToNine + oneToNine + tenToNineteen;

    hundredToThousand =
(9*100*letters[27].length())+(99*9*and.length())+(9*total)+(oneToNine*100);
total = total+letters[28].length()+letters[0].length()+hundredToThousand;

    System.out.println(total)
```

Premananth Vishnugaran
2018304
Module: 5COSC007C.1