Wallace Wahong’o
2003944 BSD
Assignment II

Question 1

{10 Marks}

The Interim Independent Electoral Commission (IIEC) of Kenya requires an Electronic Voting Management System (EVMS) through which new voters can be added, invalid voters deleted as well as displaying of a voter’s details, among other functions.

Voter details include voter card ID, National ID Number, First Name, middle name, surname, polling station, date of birth (entered in dd-mm-yyyy format) and gender .

In an interview for an advertised Software Developer position, the commission has requested you to demonstrate with a simple program how a voter details can be added and displayed. Using your own defined and appropriate voter class and an interactive driver program (i.e. main function) write a C++ program to achieve this.

Voter data class

```
public class Voter
{
   public string First_Name { get; set; }

   public string Last_Name { get; set; }

   public int ID_Number { get; set; }

   public DateTime Date_Of_Birth { get; set; }

   public string Polling_Station { get; set; }

   public string Gender { get; set; }

}
```

Voter management class

it manages voters registration, display voters

```
public class VoterManagement
{
   public string Polling_Station { get; set; }

   Console.WriteLine("Enter Polling_Station: ");

   try
   {
    Polling_Station = Console.ReadLine() ?? throw new ArgumentNullException();
}
catch (System.Exception)
{
    Console.WriteLine("Invalid Polling_Station. Please enter a valid Polling_Station.");
    throw;
}



try
{
    voter.Polling_Station = Console.ReadLine() ?? throw new ArgumentNullException();
}
catch (System.Exception)
{
    Console.WriteLine("Invalid Polling_Station. Please enter a valid Polling_Station.");
    throw;
}

string path = "voters.json";

public List<Voter> LoadVoters()
{
    List<Voter> votersList = new List<Voter>();

    if (File.Exists(path))
    {
        string [] lines = File.ReadAllLines(path);

        foreach (string line in lines)
        {
            Voter voter = JsonSerializer.Deserialize<Voter>(line);
            votersList.Add(voter);
        }

        return votersList;
    }
    else
    {
        Console.WriteLine("No voters found.");
        return votersList;
    }

}


public void RegisterVoter()
{
```

```
Voter voter = new Voter();

Console.WriteLine("Register Voter... \n\n");

//get voter details
Console.WriteLine("Enter first name: ");
try
{
    voter.First_Name = Console.ReadLine() ?? throw new ArgumentNullException();
}
catch (System.Exception)
{
    Console.WriteLine("Invalid name. Please enter a valid name.");
    throw;
}

Console.WriteLine("Enter last name: ");
try
{
    voter.Last_Name = Console.ReadLine() ?? throw new ArgumentNullException();
}
catch (System.Exception)
{
    Console.WriteLine("Invalid name. Please enter a valid name.");
    throw;
}

Console.WriteLine("Enter your ID number: ");

try
{
    voter.ID_Number = Convert.ToInt32(Console.ReadLine());
}
catch (System.Exception)
{
    Console.WriteLine("Invalid ID number. Please enter a valid ID number.");
    throw;
}


voter.Date_Of_Birth = DateTime.Now;

List<Voter> voters = LoadVoters();

if (voters.Exists(v => v.ID_Number == voter.ID_Number))
{
    Console.WriteLine("Voter already exists.");
    return;
}
```

```
        string voterJson = JsonSerializer.Serialize<Voter>(voter, new JsonSerializerOptions
{ WriteIndented = true });

        File.AppendAllText("voters.json", voterJson + Environment.NewLine);
    }


    public void DisplayVoters()
    {
        var voters = LoadVoters();

        foreach (var voter in voters)
        {
            Console.WriteLine($"Name: {voter.Name}, Age: {voter.Age}, ID Number:
{voter.ID_Number}");
        }
    }

}
```

voter when registered, the class will be converted to JSON and added to a file called voters.json. To display the voters, the json data will be deserialized and converted to Voter class.

Voter System Class

```
public class VoterSystem()
{
    public void Run()
    {
        VoterManagement voterManagement = new VoterManagement();

        Console.WriteLine("Voter System");

        var consoleActive = true;
        while (consoleActive)
        {

            Console.WriteLine("1. Register Voter");
            Console.WriteLine("2. Display Voters");
            Console.WriteLine("3. Exit");

            Console.WriteLine("Enter your choice: ");
            int choice = Convert.ToInt32(Console.ReadLine());

            switch (choice)
            {
                case 1:
                    voterManagement.RegisterVoter();
                    break;
                case 2:
                    voterManagement.DisplayVoters();
                    break;
                case 3:
                    Environment.Exit(0);
                    consoleActive = false;
                    break;

                default:
                    Console.WriteLine("Invalid choice. Please enter a valid choice.");
                    break;
            }
        }

    }
}
```

Program Class

```
class Program
{
    static void Main(string[] args)
    {

        VoterSystem voterSystem = new VoterSystem();
        voterSystem.Run();
    }
}
```

Question 2
{10 Marks}

Safaricom Ltd intends to develop an Electronic Reward system aimed at rewarding its subscribers with Bonus (“Bonga”) points, where subscribers can redeem points for airtime or prizes. The system shall track the amount of air time a subscriber has and reward as follows:

Airtime
Bonus Points
Equal or above Ksh. 2000.00
500
Ksh. 1,000.00 - Ksh 1,999.00
300
Ksh. 500.000 - Ksh 999.00
100
Ksh. 100.000 - Ksh 499.00
50
Below Ksh. 100.00
0

Write a C++ program that captures the Subscriber name, Phone number and Air time amount, through a constructor, uses a function compute_bonuspoints() to calculate the points awarded, then outputs information as follows:

```
namespace Bonga_reward;

public class Subscriber
{
    public string Name { get; set; }
    public string PhoneNumber { get; set; }
    public double AirtimeAmount { get; set; }

    public Subscriber(string name, string phoneNumber, double airtimeAmount)
    {
        Name = name;
        PhoneNumber = phoneNumber;
        AirtimeAmount = airtimeAmount;
    }

    public int ComputeBonusPoints()
    {
        if (AirtimeAmount >= 2000)
        {
            return 500;
        }
        else if (AirtimeAmount >= 1000)
        {
            return 300;
        }
        else if (AirtimeAmount >= 500)
        {
            return 100;
        }
        else if (AirtimeAmount >= 100)
        {
            return 50;
        }
        else
        {
            return 0;
        }
    }

    public void DisplayInformation()
    {
        Console.WriteLine($"{Name}: (PHONE NO: {PhoneNumber}):  AWARDED {ComputeBonusPoints()} bonus points, STAY WITH SAFARICOM. THE BETTER OPTION.");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Subscriber subscriber = new Subscriber("Wallace Wahongo", "0712345678", 1500);
        subscriber.DisplayInformation();
    }
}
```