Object Oriented Programming, Jan-May 2023
Mid-Semester Exam (100 points) 1 HOUR 30 MINS

Date: Friday, March 3, 2023

Name: **Student ID Number:**

Instructions:

- Answer **ALL** questions.
- The Ashesi Honour Code applies here.

Good Luck!

Problem 1 [20 points]

Flow of Control

a) [8 points] Name the key parts of a loop.

b) [12 points] Step through the following code segment.

```
Scanner input = new Scanner(system.in);
int usersNumber = input.nextInt();
int numFactors = 0;
int factorToCheck = 2;

while (factorToCheck < usersNumber) {

      if (usersNumber % factorToCheck == 0) {
            numFactors++;
      }

      factorToCheck++;
}

if (numFactors == 0)
      System.out.println(usersNumber + “ is prime.”);
else
      System.out.println(usersNumber + “ is not prime.”);
```

(ii) [6 points] Assume the user enters **5** as input. How many times will the loop run? What will be the value of numFactors after the loop? What will be the output of the programme?

(iii) [6 points] Assume the user enters **6** as input. How many times will the loop run? What will be the value of numFactors after the loop? What will be the output of the programme?

Problem 2 [25 points]
Elections I: Object Oriented Design, Java Classes

Generally, an election is a process in which qualified people vote to choose a person or group of people to hold an official position. For the next several questions, we will consider a presidential election; think of the very recent Nigerian election. The class diagram (i.e., UML - Object Oriented Design) below provides an overview of a `Citizen` class:

```
Citizen
---------------------------------------------------
- citizenName                    : String
- citizenAge                     : int
- countryOfCitizenship           : String
---------------------------------------------------
+ getCitizenName()
+ getCitizenAge()
+ getCountryOfCitizenship()
+ setCitizenName(String name)
+ setCitizenAge(int citizenAge)
+ setCountryOfCitizenship(String citizenship)

+ equals(Citizen other)
+ toString()
+ hashCode()
```

a. [12 points] Begin your task by implementing the `Citizen` class focusing on the:

   I. [4 points] Header and fields (i.e., member/instance variables) of the class (see above).

   II. [8 points] A constructor that takes in citizen's *name*, *age* and *country of citizenship* as parameters. Do not allow the age to be set to a negative number. Rather, if a negative number is provided, set the age to 0 (representing a baby).

b. [8 points] Implement the `equals` method for the Citizen class.

c. [5 points] Briefly explain what is meant by *method overloading.* How could I apply that concept to the constructor of the Citizen class?

Problem 3 [30 points]
Elections II: Java Tester Classes, Object Reference, Arrays

a. [4 points] Briefly explain the difference between an instance variable and a local variable.

b. [4 points] Briefly explain the difference between a method that is static and one that is not static.

c. [8 points] Create a `TestCitizen` class (i.e. driver class) that contains a main method for testing the Citizen class from **Problem 2** above:

   I. Instantiate four (4) `Citizen` objects using the details in the table below:

Object Reference	Name	Age	Country of Citizenship
`citizenTinubu`	Tinubu	85	Nigeria
`citizenOlympio`	Olympio	50	Togo
`citizenObi`	Obidience	15	Nigeria
`citizenAtiku`	Atikulate	70	Nigeria

II. create `citizenArr`, an array of Citizens that stores all the `Citizen` objects created above.

d. [8 points] Draw the variables and objects in the computer's memory after the code for part I and II above has been executed.

e. [2 points] What happens when one of the object references is assigned to the other in the manner below?

```
citizenObi = citizenTinubu;
```

f. [4 points] Explain what happens in memory when the following expressions are being evaluated

```
I.    citizenObi == citizenTinubu;
II.   citizenObi.equals(citizenTinubu);
```

Problem 4 [25 points]

Java Classes, Control Flow

Below is the beginning of the implementation of an Election class. It shows the member variables of the class as well as the implementation of the constructor of the class. Notice that the class holds an array of Citizen objects who are the candidates for the election. It also has an array of integers which represents the number of votes each candidate has received. Please note that you can assume that all methods of the Citizen class have been fully and accurately implemented.

```
public class Election {

      private String country;       // the country this election is for
      private int minCandidateAge;  // candidates must be at least this age
      private int minVotingAge;     // voters must be at least this age
      private Citizen[] candidates; // an array of candidates for this election
      private int numCandidates;    // the number of candidates registered
      private int[] votes;          // the number of votes for each candidate
      public static final int MAX_CANDIDATES;   // max allowed # of candidates

      /**
       * Constructor of the Election class
       * @param country - The country for this election
       * @param minCandidateAge - The minimum age allowed for candidates
       * @param minVotingAge - The minimum age allowed for voters
       */
```

```
public Election(String country, int minCandidateAge, int minVotingAge)
{
      this.country = country;
      this.minCandidateAge = minCandidateAge;
      this.minVotingAge = minVotingAge;

      candidates = new Citizen[MAX_CANDIDATES];
      votes = new int[MAX_CANDIDATES];
      numCandidates = 0;
}
```

For example, in Nigeria, one must be 18 years old to vote and 35 years old to run for election as president. Using the constructor above, we could create a object to represent the Nigerian presidential election as follows:

```
Election nijaPresElection = new Election("Nigeria", 35, 18)
```

a. [5 points] In the constructor defined above, what will happen if the '`this`' keyword is removed? Provide the name of the concept which defines such a situation.

b. [10 points] The ***registerCandidate*** method, of the Election class, whose header is shown below, takes a Citizen representing a candidate as a parameter and adds the citizen to the array of candidates for the election, incrementing the number of registered candidates. The candidate can be added only if (i) the candidate's country of citizenship matches the election country, (ii) the candidate is at least as old as the specified minimum candidate age for the election, and (iii) the maximum number of candidates has not been exceeded. The method returns a boolean indicating whether or not the candidate was registered successfully. Implement this method.

```
public boolean registerCandidate(Citizen candidate) {

      // to be implemented

}
```

Choose <u>one</u> out of part c and part d to answer:

c. [10 points] The ***castVote*** method allows a voter (who is a Citizen) to cast a vote for the candidate of a given name. If the voter is allowed to vote (i.e. is at least of the minimum age to vote and a citizen of the election country), this method finds the index of the desired candidate in the array. If found, it should then increment the number of votes for that candidate by incrementing the corresponding index of the votes array. For example, if the specified candidate is found in index 3 of the candidates array, then the value in index 3 of the votes array should be incremented. The method returns true if the vote was cast successfully, and false otherwise. Implement this method.

```
public boolean castVote(Citizen voter, String candidateName) {

      // to be implemented

}
```

[Remember: Answer only <u>one</u> out of part c and part d]

d. [10 points] The ***determineWinner*** method steps through the votes array to determine the index with the highest number of votes. It then returns the **name** of the candidate corresponding to the highest number of votes. For example, if the highest value in the votes array occurs at index 4, then the name of the candidate at index 4 of the candidates array should be returned. Implement this method.

```
public String determineWinner() {

      // to be implemented

}
```

Extra Credit (5 points):

Implement the last method in Problem 4 (either c or d) which you chose not to answer earlier.