(University of Choice)

MASINDE MULIRO UNIVERSITY OF SCIENCE AND TECHNOLOGY (MMUST)

MAIN CAMPUS

UNIVERSITY MAIN EXAMINATIONS 2023/2024 ACADEMIC YEAR

SECOND YEAR 1ST SEMESTER EXAMINATIONS

BACHELOR OF SCIENCE IN COMPUTER SCIENCE, INFORMATION TECHNOLOGY, EDUCATION TECHNOLOGY, MATHEMATICS, STATISTICS & EDUCATION SCIENCE

COURSE CODE: BCS 210 / BIT 210
COURSE TITLE: OBJECT-ORIENTED PROGRAMMING

DATE: Monday 11TH December 2023 **TIME:** 8:00AM - 10:00AM

INSTRUCTIONS TO CANDIDATES

THIS IS AN OPEN BOOK EXAMINATION
Answer Question **ONE (1)** and Any **OTHER 2** questions
Ensure your answers/ideas are clearly expressed
All your answers must be clearly numbered
Write in ink. Rough work can be done in pencil and will not be marked. Cross out any rough work.
Calculators, phones, tablets, computers not allowed

TIME: 2 Hours 20 Minutes (20 minutes for reading and choosing questions)

MMUST observes ZERO tolerance to examination cheating

This Paper Consists of 9 Printed Pages. Please Turn Over. ▶

QUESTION ONE: COMPULSORY QUESTION [30 MARKS]

(a) A random password contains two upper case alphabets, followed by two digits and four lower case alphabets. Write a Java piece of code to generate this password and display it. [4 Marks]

(b) The **E**xpected **D**ate of **D**elivery (**EDD**) for an expectant woman is 282 days from the day she conceived. Given a partial class definition, where *day, month and year* hold the date the woman conceived

```
class EDD
{
   private static int days[] = {0, 31, 28, 31, 30, 31, 30, 31,
  31, 30, 31, 30, 31};
 private static final String[] week = {"Mo", "Tu", "We", "Th", "
  Fr", "Sa", "Su"};
 private static final String[] monthsNames = new String[]{"
  January", "February", "March", "April",
         "May", "June", "July", "August", "September", "
  October", "November", "December"};
   private int year;
   private int month;
   private int day;
}
```

Note that February has 28 days during a regular year and 29 days during a leap year.. The index of the day on which 1^{st} of January falls is given by the formula

$$i = R(5((year-1),4)+4((year-1),100)+6((year-1),400),7)$$

where R(x,y) is a function that returns remainder after dividing x by y.

(i) Write the set and get methods for the **EDD** class. Ensure day and month are validly initialized and year is initialized to a value between 2000 and 2100. [6 Marks]

(ii) Write a method to calculate the index of the day of which first January of a given year falls, such that this method must be invoked using a reference variable of the class EDD. [3 Marks]

(iii) A leap year is divisible by 400 or divisible by 4 and not divisible by 100. Write a method that changes the number of days in February to 29 if a year is a leap year, otherwise it number of days in February to 28. [3 Marks]

(iv) Write a method that return the name of the day on which the woman is expected to deliver. [4 Marks]

(v) Using a **while** loop within a **for** loop, write a method **public void calendar()** that displays the calendar of the month the woman will deliver. If she is expected to deliver in December 2023, your function should display the output below [10 Marks]

Example Output

```
December 2023
Mo  Tu  We  Th  Fr  Sa  Su
27  28  29  30   1   2   3
 4   5   6   7   8   9  10
11  12  13  14  15  16  17
18  19  20  21  22  23  24
25  26  27  28  29  30  31
1   2   3   4   5   6   7
```

QUESTION TWO [15 MARKS]

(a) You can estimate the value of π using the series

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \ldots\ldots \frac{4}{2n-1}$$

were n is the number of times you improve the estimation. Using a **for** loop, write a Java code fragment that estimate the value of π with $n = 20$

[4 Marks]

(b) Figure 1 shows the structure of an array in memory named **vals**.

34	56	76	12	54	75.1	48
56	71	90.25	39			
28	63.01	48	32	65		
82	36.2	84	67	75	65	
60	83.3	54				

Figure 1: Structure of an array in memory

(i) Write Java lines of code that could have been used to declare, create and initialize this array. [3 Marks]

(ii) Without using a loop, write a Java line of code that will copy first 3 elements of the first row of this array into an array declared as [2 Marks]

```
int data [] = new int[10];
```

(iii) Using a **while** loop within a **do...while** loop, write a method that will receive such an array, calculate and return mean of the values in such an array. [4 Marks]

(c) With illustration, explain why you would prefer defining an **Interface** rather than an **abstract** class. [2 Marks]

QUESTION THREE [15 MARKS]

Consider partial class definitions for **Shape2D** and **Circle** classes

```
public abstract class Shape2D {
 private String name;
  private Date doc;

 public Shape2D(String name){
  this.name = name;
  doc = new Date();
 }
 public abstract double area();
 public abstract double perimeter();
}
class Circle extends Shape2D{
 private double radius;

 @Override
 private double perimeter(){
  return 2 * radius * Math.PI;
 }
}
```

(i) Identify and explain **TWO** errors in these class definitions. [2 Marks]

(ii) Write the missing two lines of code of a **Circle** constructor

```
public Circle(double radius, String name){
   ----
   ----
 }
```

such that it initializes radius and name of the shape with parameters it receives [2 Marks]

(iii) Assuming the method **area()** is properly overridden in class **Circle**, explain the error in line 2 of code below.Note that line 1 is very correct. [2 Marks]

```
1 Object o = new Circle(2.5, "Circle");
2 o.area();
```

(iv) Re-write line 2 of code in (iii) above such that the error identified is corrected. [2 Marks]

(v) The default implementation of the **equals** method in the **Object** class is

```
public boolean equals(Object obj) {
  return (this == obj);
}
```

This implementation checks whether two reference variables point to the same object using the == operator. Write the definition of the overridden method **equals** as it should appear in class **Circle** so that it checks if two **Circle** objects have equal radii. [4 Marks]

(vi) Assume abstract class **Shape2D** has a method defined as

```
public Date getDOC(){
  return doc;
}
```

and class **Circle** has no public set methods. Is the class **Circle** immutable? Explain your answer. [3 Marks]

QUESTION FOUR [15 MARKS]

Taylor series for **sin(x)** and **cos(x)** are defined as

$$sin(x) = \sum_{n=1}^{\infty}(-1)^{n+1}\frac{x^{2n-1}}{(2n-1)!}$$

$$cos(x) = \sum_{n=0}^{\infty}(-1)^{n}\frac{x^{2n}}{(2n)!}$$

where x is the angle in radians and n is the number of repetitions. Given an angle in degrees (**deg**), we can convert it into radians (**rad**) using the formula

$$rad = \frac{deg}{180} \times \pi$$

where π is a constant whose value is approximately 3.14159. Given a partial class definition as

```
class Trig {
  private static double x;
}
```

Given that

$$tan(x) = \frac{sin(x)}{cos(x)}$$

(i) Write the definition of the set method for instance variable **x**. Valid values for **x** range from 0 to 180. [1 Marks]

(ii) Write the definition of the get method for instance variable **x** such that it can be invoked without using a reference variable of **Trig**. [1 Marks]

(iii) Write definition a method that receives an angle in degrees and return the equivalent angle in radians. This method should only be accessible within the class **Trig** [2 Marks]

(iv) Write definition of a recursive method that receives an integer and returns the factorial of that integer [2 Marks]

(v) Using a **while** loop, write definition of a method named **sinx** which calculates and returns the sin of an angle x. Re-use methods defined in (ii), (iii) and (iv) above. Let n vary from $0 \ldots 10$ [2 Marks]

(vi) Using a **for** loop, write definition of a method named **cosx** which calculates and returns the cosine of an angle x. Re-use methods defined in (ii), (iii) and (iv) above. Let n vary from $0 \ldots 10$ [2 Marks]

(vii) Write definition of a method named **tanx** which calculates and returns the tangent of an angle x. Re-use methods defined in (v) and (vi). [2 Marks]

(viii) Write definition of the overridden method **toString()** such that it returns the sin and cosine of an angle **x**. For example, if x is initialized to 30^o, this method should return sample output shown below [3 Marks]

Example Output

```
Ange in degrees = 30.0
Angle in radians : 0.5235987755982988
cos(30.0) = 0.8660254037843734
sin(30.0) = 0.4999999999999482
tan(30.0) = 0.5773502691896094
```