SOFTWARE APPLICATION DEVELOPMENT

SECTION A

Answer ALL questions (24 marks)

Please answer the following true/false questions. give a brief explanation of your answer. [10 marks].

a. Class and object mean the same thing.
b. An interface can be used as the type of a variable.
c. A method signature includes information about the number and types of parameters that a method has.
d. To be notified when a user clicks on a button, the program must add a listener to the button.
e. A subclass can add behaviour that is not present in the superclass.
f. If FileNotFoundException occurs when a program is executed, it means that the program contains a bug.
g. Instance variables should always be declared to be public.
h. Variables should be declared to be local, rather than instance variables, whenever possible. (If the program would not work if they were local variables, consider that to be "not possible".)
i. Calling nextInt() on a Scanner object two times in a row will always return the same value.
j. An exception can be thrown in one method and handled in the method that called it.

1. Write a java program to print the following patterns using for loops.

```
1010101

 10101

  101

   1
```

i. Write a class called Room, which has three private instance variables: a double width, representing the width of the room in feet. a double length, representing the length of the room in feet, and an int floor, representing the building floor that the room is on. [2 marks].

ii. Write a default constructor for the class Room that sets the width to 10, the length to 12.5, and the floor to 1. [1 mark].

iii. Write get and set methods (getters and setters) for the three instance variables. For the set methods for the width and length. only positive values should be set. If the input is 0 or a negative number, the variables should not be changed. [2 marks].

iv. Write a constructor for the class Room that takes in two double parameters and an int, and sets length to the larger double. width to the smaller double, and floor to the int. Use the setters from part (b). [1 mark].

v. Override the default toString method for the class Room, so that it a String with the form "length x width on floor floor". For example, calling toString on the instance of Room created by the default constructor from part (a) would return the String "12.5 x 10, floor 1" [1 mark].

vi. Write a subclass of the class Room called Classroom, which also has a private instance variable of type int called numStudents. representing the maximum number of students that the classroom can hold. [1 mark].

vii. Write a constructor for Classroom which takes in two double variables and two ints. The instance variable length should be set to the larger double, and the width should be set to the smaller double as in Room. The instance variable floor should be set to the first int and the instance variable numStudents should be set to the second int. Leave the instance variable as private in Room and use setters to access them. [1 mark].

viii. Write a toString method for Classroom which uses the toString method for Room, followed by an additional String ", capacity = numStudents students", where numStudents is replaced by the instance variable value

ix. Write a static method which takes in an array of Rooms and returns an array of Classrooms that contains exactly those Rooms in the input array that are also Classrooms.

SECTION B (36 marks)

4. a. The Java class called Song is started below. An object of class Song is a piece of music. This class has four instance variables:

- title, which is a String representing the title of the song
- artist, which is a String representing the performer of the song.
- length, which is an int representing the length of the song in seconds.
- composer, which is a String, representing the composer of the song.

i. Write a constructor which takes in an artist, length, title and composer, and instantiates an object Song with those values.

ii. Write a nonstatic method artistsComposer which returns true if the artist and the composer are the same for the Song, and false otherwise.

iii. Use your constructor from part (a) to instantiate an object of class Song with title "Halo", artist "Beyonce", composer "Ryan Tedder", and length 261 seconds.

iv. Write a static method totalt.ength, which takes in an array of Song objects, and returns the sum of the lengths of the songs in the array. You may assume that every element of the array contains a Song object

   - Write a static method called swap that swaps two elements of an int array. The method should accept three input parameters:
   - an int representing the index of one element being swapped.
   - an int representing the index of the other element being swapped.
   - an int array the method does not return anything.

b. Write a static method which takes an array of ints as its argument, and reverses the order of the elements in the array. That is, the first and last elements are switched, the second and second last elements are switched, etc. For example, this method should change the array [1. 2. 3, 4] to the array 14, 3, 2, 11.

Be Humble…………………………

1. A Rectangle can be implemented as a class with two instance variables: length and width Write NegativeValueException which extends a class Exceptions and has two constructors in which first constructor does not have any parameter and second constructor which has string message parameter. Create a constructor which has default values set to 1. Create a second constructor which pass length and width and throws NegativeValueException if length or width is less than 0. Provide a set function which set both length and width and throws an exception if length and width is less than 0. Provide getArea function which returns area of rectangle an getPerimeter functions which returns perimeter of rectangle.

2. Do the following for the subclass Dog: [12 marks]

   a. Define the default constructor so it sets has Tail to true, numLegs to 4, and mass to 10.0 as in the parent class.
   b. Create a setter for mass that throws an Exception if the input parameter is less than 2 or greater than 350.
   c. Write an equals method for Dog. It should return true if the breed and name of the two Dog instances are the same, and false otherwise.

3. The interface GCD contains an abstract method computeGCD(int num1, int num2). Class APPROACH1 implements the interface by following Euclid's algorithm and class APPROACH2 implements the interface by listing all the factors (need not be prime factors) of the two numbers and choosing the highest common factor. Write a Java program to do the above said operations.

4. Write a program to prompts the user to enter a length in feet and inches and outputs the equivalent length in centimetres. if the user enters a negative number or a non-digit number, throw and handle an appropriate exception and prompt the user to enter another set of numbers

5. **Question**

i. Create a super class called Car. The Car class has the following fields and methods.:

- intspeed;
- doubleregularPrice;
- Stringcolor;
- doublegetSalePrice();

ii. Create a sub class of Car class and name it as Truck. The Truck class has the following fields and methods:

- intweight;
- doublegetSalePrice(); //Ifweight>2000,10%discount.Otherwise,20%discount.

iii. Create a subclass of Car class and name it as Ford. The Ford class has the following fields and methods:

- intyear;
- ntmanufacturerDiscount;
- doublegetSalePrice();//FromthesalepricecomputedfromCarclass,subtractthema nuf acturerDiscount.

iv. Create a subclass of Car class and name it as Sedan. The Sedan class has the following fields and methods.

- intlength;
- doublegetSalePrice();//Iflength>20feet,5%discount,Otherwise,10%discount.

v. Create MyOwnAutoShop class which contains the main() method. Perform the following within the main() method.

   a. Create an instance of Sedan class and initialize all the fields with appropriate values. Use super(...) method in the constructor for initializing the fields of the superclass.
   b. Create two instances of the Ford class and initialize all the fields with appropriate values. Use super(...) method in the constructor for initializing the fields of the super class.
   c. Create an instance of Car class and initialize all the fields with appropriate values. Display the sale prices of all instance

6. Given the following multi-dimension array initialization.

```
int my array[][] = {{1,2,4}, {2,3,0], {5,6,1}};
```

Write a java program to print sum of every column.

Expected output:

```
Sum of column1: 8
Sum of column2: 11
Sum of column3: 5
```

7. Write a program that uses nested loops to collect data and calculate the average rainfall over a period of years. The program should first ask for the number of years. The outer loop will iterate once for each year. The inner loop will iterate twelve times, once for each month. Each iteration of the inner loop will ask the user for the inches of rainfall for that month. After all iterations, the program should display the number of months, the total inches of rainfall, and the average rainfall per month for the entire period.

8. Write a program with a loop that asks the user to enter a series of positive numbers. The user should enter a negative number to signal the end of the series. After all the positive numbers have been entered, the program should display their sum.

9. Write a Java application that inputs a series of integers and determines and prints the largest integer. Your program should use at least the following three variables: (a) counter: A counter to count to 10 (i.e., to keep track of how many numbers have been input and to determine when all 10 numbers have been processed). (b) number: The integer most recently input by the user. (c) largest: The largest number found so far.

10. Write a program which prompts user for the number of students in a class, It then prompts user for the grade of each of the students (integer between 0 to 100) and saves them in an int array called score. The program shall then compute and print the average (in double rounded to 2 decimal places) and minimum/maximum (in int).

11. Write a program that asks the user to enter the amount that he or she has budgeted for a month. A loop should then prompt the user to enter each of his or her expenses for the month, and keep a running total. When the loop finishes, the program should display the amount that the user is over or under budget.

12. Write a Java program to d store the score into an array and output the stored score on the computer screen descending order (from largest to smallest). knowing that the number of the students and the student score shall be given by the user.

13. Consider the following code fragment:

```
int sum = 0;
int i = 0;
while (i < 5)  {
         sum += i;
        i++;
        }
System.out.print(sum);
```

Replace the while loop in the fragment above with a for loop that prints the same value of sum variable.

14. A certain CS professor gives 5-point quizzes that are graded on the scale 5-A, 4-B, 3-C, 2-D, 1-F, 0-F. Write a program that accepts a quiz score as an input and uses a decision structure to calculate the corresponding grade.

15. Write Java program to allow the user to input two integer values and then the program prints the results of adding, subtracting, multiplying, and dividing among the two values. See the example below: Enter value a:30 Enter value b:10 The result of adding is 40. The result of subtracting is 20; The result of multiplying is 300. The result of dividing is 3

16. . Write a program to display all odd numbers between 0 and 1000.

17. Practical Test 1 – Dec 29/12/2022 10:50-11:20 AM (CHWAKA)

Question 1:

Write a class for a video game character. The character should have a name, a type (scout, soldier, medic, etc.) and current health. Therefore, it needs three attributes:

•String name

•String type

•int health

This class should have the following methods:

•GameCharacter(String newName, String newType, newCurHealth)

—Constructor that takes three inputs

•changeHealth(int change)

—A method that changes the health of the character. The character's health will change by change amount, so it will go down if change is negative, and up if it's positive. If the health goes below 0, changeHealth should return the String "Your character is dead".

Name this class GameCharacter. Write a main method to test your code. Each method should be tested at least 3 times.

Question 2

Consider a class that keeps track of the sales of an item. An object of this class will have the attributes (variable names in parentheses):

•Number sold (numSold)

•Total sales in dollars (totSales)

•Total discounts (totDisc)

•Cost per item (itemCost)

•Bulk quantity: the number that must be ordered at once to trigger a bulk discount (bulkQuant)

•Bulk discount percentage: in a bulk order, the percentage discount—i.e. a value of .3 would mean 30% off (bulkDiscPct)

And the following constructors:

•ItemSales()

—Default constructor.

•ItemSales(double itemCost)

—Constructor that initializes the itemCost.

Implement this class in Java. (What types should the variables be? What values should the variables be initialized to in each contructor?) Implement get and set methods for all attributes, as well as a toString method. Now write a main method to test your setters and getters. Make sure you test all the possibilities (e.g., what happens if you try to set a negative cost per item?). Also be sure the output is clear for your users.

Name these classes ItemSales & ItemSalesDemo.

Question 3

PersonAddress Class Implement a class PersonAddress that represents an entry in an address book. Its attributes are:

•The first name of the person

•The last name of the person

•The email address of the person

•The telephone number of the person

It will have methods to

•Access each attribute

•Change the email address

•Change the telephone number

•Test whether two instances are equal based solely on name

- •Return a String with the information stored in that instance of the PersonAddress class (a toString() method).

Include a driver program that instantiates at least three objects of type PersonAddress and calls each method at least twice.

Question 4

Consider a class Characteristic that will be used in an online dating service to assess how compatible two people are. Its attributes are:

- •description—A String that identifies the characteristic (for instance pets or hair color)
- •rating—An integer between 1 and 10 (inclusive) that indicates a person's desire for this characteristic in another person

And the following methods:

- •Write a constructor that sets the description of the characteristic to given String and sets the rating to zero to indicate that it has not yet been determined.
- •Write a private method isValid(aRating) that returns true if the given rating is valid, that is, it is between 1 and 10
- •Write a method setRating(aRating) that sets the rating to aRating if it is valid
- •Write a method setRating() that reads a rating from the keyboard. If the rating entered is not valid, keep asking the user until they enter one that is valid.
- •Implement get, and set methods for all attributes (except the setter for rating, which was done above).

Implement this class in Java and write a main() method within it to test it

18. Define a method named 'perfect' that determines if parameter number is a perfect number. Use this function in a program that determines and prints all the perfect numbers between 1 and 1000
Hint: An integer number is said to be "perfect number" if its factors, including 1 (but not the number itself), sum to the number. Eg, 6 is a perfect number because 6 = 1+2+3.

19. What is the difference between Exceptions and Errors?
20. What is the purpose of Exceptions? Give an example of what they're used for.
21. With the examples briefly explain the two forms of polymorphism in java.

22. Write a do-while loop that asks the user to enter two numbers. The numbers should be added and display the sum. The loop should ask the user whether he or she wishes to perform the operation again If so, the loop should repeat, otherwise it should terminate.

23. Write a program that takes your full name as input and displays the abbreviations of the first and middle names except the last name which displayed as it is. For example if your name is Robert Brett Roser , then the output should be R.B.Roser.

24.
```
int a = 6;
int b = a/4;
int c = a % 4;
float d = a / 4.0f;

System.out.println("a = "+ (a + b) +" + b + c = "+ c + " d = "+d);
```

25.
```
for(int i = 0; i <= 10; i = i+2){

        System.out.print(i+ ":"+ i+3+"-");

}//end of for

System.out.println();
System.out.println("The End");
```

26. Create a class called Book to represent a book. A Book should include four pieces of information as instance variables-a book name, an ISBN number, an author name and a publisher. Your class should have a constructor that initializes the four instance variables. Provide a mutator method and accessor method (query method) for each instance variable. Inaddition, provide a method named getBookInfo that returns the description of the book as a String (the description should include all the information about the book). You should use this keyword in member methods and constructor. Write a test application named BookTest to create an array of object for 30 elements for class Book to demonstrate the class Book's capabilities.

27. (Date Class) Create a class called Date that includes three pieces of information as data members a month (type int), a day (type int) and a year (type int). Your class should have a constructor with three parameters that uses the parameters to initialize the three data members. For the purpose of this exercise, assume that the values provided for the year and day are correct, but ensure that the month value is in the range 1-12; if it is not, set the month to 1. Provide a set and a get function for each data member. Provide a member function displayDate that displays the month, day and year separated by forward slashes (/). Write a test program that demonstrates class Date's capabilities.

28. (Employee Class) Create a class called Employee that includes three pieces of information as data members a first name (type string), a last name (type string) and a monthly salary (type int). Your class should have a constructor that initializes the three data members. Provide a set and a get function for each data member. If the monthly salary is not positive, set it to 0. Write a test program that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10 percent raise and display each Employee's yearly salary again.

29. Create a java class called Dog with two instance variables name (typeString) and age (typeInt) then Define a constructor that will initialize these variables and provide a getter and setter methods for both name and age. Finally write a program that demonstrates the creation of a Dog objects, setting its attributes and display them.

30. (Invoice Class) Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as data members a part number (type string), a part description (type string), a quantity of the item being purchased (type int) and a price per item (type int). Your class should have a constructor that initializes the four data members. Provide a set and a get function for each data member. In addition, provide a member function named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as an int value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0. Write a test program that demonstrates class Invoice's capabilities.

31. Create a class called Car that includes three instance variables:

- A model (type String)
- A year (type String)
- A price (double).

Provide a constructor that initializes the three instance variables then provide a set and a get method for each instance variable. If the price is not positive, do not set its values. Write a test application named CarApplication that demonstrates class Car's capabilities. Create two Car objects and display each object's price. Then apply a 5% discount on the price of the first car and a 7% discount on the price of the second. Display each Car's price again.

32. (Rectangle Class) Create a class Rectangle with attributes length and width, each of which defaults to 1. Provide member functions that calculate the perimeter and the area of the rectangle. Also, provide set and get functions for the length and width attributes. The set functions should verify that length and width are each floating-point numbers larger than 0.0 and less than 20.0

33. Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has ondeposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12 this interest should be added to savingsBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value. Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of $2000.00 and $3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.

34. Create a class called Kitten that has three fields: String name, String owner, and int age. Create a constructor for Kitten that takes a String name and owner for the Kitten and uses them for initialization. Have the age for a Kitten start at 0; Implement a method for age which return a Kitten's age. Implement a method called haveBirthday that does not return anything and simply increases a Kitten's age by one. Finally, write a method called toString that returns a string of the form:" is and belongs to" e.g." Bob is 87 and belongs to Gregor Samsa". Create an object which initialize three data members and print out the string above.

35. (HugeInteger Class) Create a class HugeInteger that uses a 40-element array of digits to store integers as large as 40 digits each. Provide member functions input, output, add and substract. For comparing HugeInteger objects, provide functions isEqualTo, isNotEqualTo, isGreaterThan, isLessThan, isGreaterThanOrEqualTo and isLessThanOrEqualToeach of these is a "predicate" function that simply returns TRue if the relationship holds between the two HugeIntegers and returns false if the relationship does not hold. Also, provide a predicate function isZero. If you feel ambitious, provide member functions multiply, divide and modulus.

36. (Complex Class) Create a class called Complex for performing arithmetic with complex numbers.
Write a program to test your class. Complex numbers have the form realPart + imaginaryPart * I where i is Use double variables to represent the private data of the class. Provide a constructor that enables an object of this class to be initialized when it is declared. The constructor should contain default values in case no initializers are provided. Provide public member functions that perform the following tasks:

    a. Adding two Complex numbers: The real parts are added together and the imaginary parts are added together.
    b. Subtracting two Complex numbers: The real part of the right operand is subtracted from the real part of the left operand, and the imaginary part of the right operand is subtracted from the imaginary part of the left operand.
    c. Printing Complex numbers in the form (a, b), where a is the real part and b is the imaginary part.

37. (Rational Class) Create a class called Rational for performing arithmetic with fractions. Write a program to test your class. Use integer variables to represent the private data of the class the numerator and the denominator. Provide a constructor that enables an object of this class to be initialized when it is declared. The constructor should contain default values in case no initializers are provided and should store the fraction in reduced form. For example, the fraction would be stored in the object as 1 in the numerator and 2 in the denominator. Provide public member functions that perform each of the following tasks:

    a. Adding two Rational numbers. The result should be stored in reduced form.
    b. Subtracting two Rational numbers. The result should be stored in reduced form.
    c. Multiplying two Rational numbers. The result should be stored in reduced form.
    d. Dividing two Rational numbers. The result should be stored in reduced form.
    e. Printing Rational numbers in the form a/b, where a is the numerator and b is the denominator.
    f. Printing Rational numbers in floating-point format.

38. Define a class called Triangle with three integer data members a, b, and c as the lengths of its three edges. This class should have the following methods:

(a)a constructor with 3 parameters representing the 3 edges

(b)a method isTriangle() which returns true if the 3 edges are all positive and they satisfy the triangle inequality where a+b > c, a+c > b, b+c > a.

(c)a method getArea() which returns the area of triangle. NB: Area=1/2(a*b) The signature of these methods are given below:

```
public Triangle(int newa, int newb, int newc)
public boolean isTriangle()
public double getArea()
```

Note: getAngle() should return zero if the triangle is not really a triangle.

39. (Account Class) Create a class called Account that a bank might use to represent customers' bank accounts. Your class should include one data member of type int to represent the account balance. Your class should provide a constructor that receives an initial balance and uses it to initialize the data member. The constructor should validate the initial balance to ensure that it is greater than or equal to 0. If not, the balance should be set to 0 and the constructor should display an error message, indicating that the initial balance was invalid. The class should provide three member functions. Member function credit should add an amount to the current balance. Member function debit should withdraw money from the Account and should ensure that the debit amount does not exceed the Account's balance. If it does, the balance should be left unchanged and the function should print a message indicating "Debit amount exceeded account balance." Member function getBalance should return the current balance. Create a program that creates two Account objects and tests the member functions of class Account.

40. Create a class named Cat. Declare two (2) instance variables: catType (String) and cutenessLvl (int). Create a public constructor without parameters. Inside the constructor, set the values of catType and cutenessLvlto "Unknown" and 3, respectively. Add another public constructor with two (2) parameters: type (String) and Ivl (int). Inside the constructor, assign type to cat Type and level to cutenessLvl. Create an int type of method named getCuteness(). This method shall return the cat's cuteness level.

41. Consider the following program and then give an output based on the question below: Assume the program is running without an error.

```
public class Myclass{

   private static int count = 0;

       private int x;

   public Myclass(int i){
       x = i;
    }

   public void incrementCount(){
       count++;
    }

    public void printX(){

       System.out.println("Value of x is: " +x);
    }

    public static void printCount(){

       System.out.println("Value of count is: " +count);
    }

    public static void main(String[]args){

           Myclass myobject1 = new Myclass(5);
           Myclass myobject2 = new Myclass(7);

              i.  myobject1.printX();
             ii.  myobject1.incrementCount();
            iii.  Myclass.incrementCount();
             iv.  myobject1.printCount();
              v.  myobject2.printCount();
             vi.  myobject2.printX();
            vii.  myobject1.setX(14);
           viii.  myobject1.incrementCount();
             ix.  myobject1.printX();
              x.  myobject1.printCount();
             xi.  myobject2.printCount();


   } }
```

42. What will be the output of the following program?

```
public class A{

    protected int x = 1;

    protected void setX(int a){
        x = a;
    }

    protected int getX(){
        return x;
    }
}

public class B extends A {
    protected int x = 3;

    public int getX(){
        setX(2);   // call superclass method to set superclass attribut
        return x; // but return attribute of subclass
    }

    public int getB(){
        return x;
    }

}



public class C{
    public static void main(String[] args){
        A a = new A();
        A b = new B();

        System.out.println(a.getX());
        System.out.println(b.getX());

        System.out.println(a.x);
        System.out.println(b.x);
    }
}
```