



THE STATE UNIVERSITY OF ZANZIBAR

SCHOOL OF COMPUTING, COMMUNICATION AND MEDIA

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

FINAL EXAMINATION

SEMESTER II

CS 1207: OBJECT ORIENTED PROGRAMMING (OOP)

Date: 20/07/2022

Time: 02:00 PM – 5:00 PM

INSTRUCTIONS

1. This paper consists of **TWO** sections, **A** and **B**, which carries 24 and 36 marks respectively.
2. Answer all questions from section A, and any **THREE** questions from section B.
3. Cellular phones and any other unauthorized materials are **NOT** allowed in the examination room.
4. **ANSWER EACH QUESTION ON SEPERATE SHEET**
5. This exam consists of nine (8) printed pages, including cover page

SECTION A (24 marks)

Answer all questions from this section.

1.
 - a. What is the difference between Exceptions and Errors? [2 marks]
 - b. What is the purpose of Exceptions? Give an example of what they're used for. [2 marks]
 - c. With examples briefly explain the two forms of polymorphism in java. [2 marks]
2.
 - a. A class has an integer data member 'i' and a method named 'printNum' to print the value of 'i'. Its subclass also has an integer data member 'j' and a method named 'printNum' to print the value of 'j'. Make an object of the subclass and use it to assign a value to 'i' and to 'j'. Now call the method 'printNum' by this object. [3 marks]
 - b. Write a do-while loop that asks the user to enter two numbers. The numbers should be added and the sum displayed. 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. [3 marks]
 - c. 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 is displayed as it is. For example, if your name is Robert Brett Roser, then the output should be R.B.Roser. [3 marks]
- 3 a. What will be the output of the following programs? [3 marks]

```
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 attrib  
            return x; } //but return attrib 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);}}}
```

```
public class a2{
public static void main(String [] args){
    try
    {   badmethod();
        System.out.print("A");
    } catch(RuntimeException ex)
    {System.out.print("B");}
    catch(Exception ex)
    {System.out.print("C");}
    finally{System.out.print("D");}
    System.out.print("E");
}

public static void badmethod()
{throw new RuntimeException();}
}
```

[3 marks]

```
c.
public class Ham {
int a = 0;
int b = 1;
public void a() {
System.out.println("Ham " + a);
public void b() {
System.out.println("Ham " + b);
public String toString() {
return "Ham " + a + " " + b;
public class Spam extends Ham {
int a = 2;
public void a() {
System.out.println("Spam " + a);
public class Yam extends Spam {
int b = 3;
public void a() {
System.out.println("Yam " + a);
public void b() {
System.out.println("Yam " + b);
public class Polymorphism3 {
public static void main (String [] args) {
Ham[] food = { new Spam(), new Yam(),
new Ham();
for (int i = 0; i < food.length; i++) {
System.out.println(food[i]);
food[i].a();
food[i].b();
System.out.println(food[i].a);
System.out.println(food[i].b);
System.out.println();
}}}
```

[3 marks]

SECTION B (36 marks)

Answer any THREE questions from this section, each question carries 12 marks.

- 4 a. 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. [4 marks]

Hint: An integer number is said to be "perfect number" if its factors, including 1 (but not the number itself), sum to the number. E.g., 6 is a perfect number because $6=1+2+3$.

- b. Given the following multi-dimension array initialization. [4 marks]

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

Write java program to print sum of every column.

Expected output:

Sum of column 1 :8

Sum of column 2: 11

Sum of column 3: 5

- c. Write a java program to print the following patterns using for loops. [4 marks]

1010101

10101

101

1

- 5 a. Create a super class called Car. The Car class has the following fields and methods. [2 marks]

- int speed;
- double regularPrice;
- String color;
- double getSalePrice();

- b. Create a sub class of Car class and name it as Truck. The Truck class has the following fields and methods. [2 marks]

- int weight;
- double getSalePrice(); // If weight > 2000, 10% discount. Otherwise, 20% discount.

- c. Create a subclass of Car class and name it as Ford. The Ford class has the following fields and methods [2 marks]
- int year;

◦ int manufacturerDiscount;
◦ double getSalePrice() // From the sale price computed from Car class, subtract the manufacturer Discount.

- d. Create a subclass of Car class and name it as Sedan. The Sedan class has the following fields and methods. [2 marks]

◦ int length;
◦ double getSalePrice() // If length > 20 feet, 5% discount, Otherwise, 10% discount.

- e. Create MyOwnAutoShop class which contains the main() method. Perform the following within the main() method. [4 marks]

◦ 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.
◦ 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.
◦ Create an instance of Car class and initialize all the fields with appropriate values.
Display the sale prices of all instance.

- 6 a. What is the purpose of Exceptions? Give an example of what they're used for? [3 marks]

- b. Write a program that prompts the user to enter a length in feet and inches and outputs the equivalent length in centimeters. 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. [4 marks]

- c. Consider class C given below. Function Integer.parseInt throws a NumberFormatException if its argument does not contain an integer. Below class C, rewrite the class so that if Integer.parseInt throws an exception, the number 1 is used. Note that Integer.parseInt is called in two places so you may need two try-statements. Don't be concerned with how one reads from the keyboard, pausing until something is typed. [5 marks]

```
public class C {  
    /** Print the sum of two integers read from the keyboard */  
    public static void main(String[] args) {  
        System.out.println("Enter a number: ");
```

```

String s;
    Read a line from the keyboard and store it in s;

    int a= Integer.parseInt(s);

    System.out.println("Enter another number: ");

    Read a line from the keyboard and store it in s;

    int b= Integer.parseInt(s);

    System.out.println("Product: " + a*b);

}
}

```

[12 mark]

7. Consider the following classes:

```

public class Computer extends Mineral {
    public void b() {
        System.out.println("Computer b");
        super.b();
    }

    public void c() {
        System.out.println("Computer c");
    }
}

public class Mineral extends Vegetable {
    public void b() {
        System.out.println("Mineral b");
        a();
    }
}

public class Animal extends Mineral {
    public void a() {
        System.out.println("Animal a");
    }

    public void c() {
        b();
        System.out.println("Animal c");
    }
}

public class Vegetable {
    public void a() {
        System.out.println("Vegetable a");
    }

    public void b() {
        System.out.println("Vegetable b");
    }
}

```

and that the following variables are defined:

```
Vegetable var1 = new Computer();
Mineral   var2 = new Animal();
Vegetable var3 = new Mineral();
Object     var4 = new Mineral();
```

What will be the output produced by the following statements?

- i. var1.a();
- ii. var1.b();
- iii. var1.c();
- iv. var2.a();
- v. var2.b();
- vi. var3.a();
- vii. var3.b();
- viii. var4.a();
- ix. ((Computer) var1).b();
- x. ((Computer) var1).c();
- xi. ((Computer) var2).c();
- xii. ((Animal) var2).c();

8. The Java class called Holiday is started below. An object of class Holiday represents a holiday during the year. This class has three instance variables:

- name, which is a String representing the name of the holiday
- day, which is an int representing the day of the month of the holiday
- month, which is a String representing the month the holiday is in

```
public class Holiday {
    private String name;
    private int day;
    private String month;
    // your code goes here
}
```

- a) Write a constructor for the class Holiday, which takes a String representing the name, an int representing the day, and a String representing the month as its arguments, and sets the class variables to these values. [3 marks]
- b) Write a method inSameMonth, which compares two instances of the class Holiday, and returns the Boolean value true if they have the same month, and false if they do not. [3 marks]
- c) Write a method avgDate which takes an array of base type Holiday as its argument, and returns a double that is the average of the day variables in the Holiday instances in the array. You may assume that the array is full (i.e. does not have any null entries). [3 marks]
- d) Write a piece of code that creates a Holiday instance with the name "Independence Day", with the day "4", and with the month "July".

STATE UNIVERSITY OF ZANZIBAR (SUZA)



SCHOOL OF COMPUTING, COMMUNICATION AND MEDIA (SCCM)

DEPARTMENT OF COMPUTER SCIENCE AND IT

END OF SEMESTER EXAMINATION

AUGUST, 2021

BACHELOR OF SCIENCE IN COMPUTER SCIENCE- BCS20

SEMESTER	II	ACADEMIC YEAR	2020/2021
COURSE CODE & TITLE	CS1207: OBJECT ORIENTED PROGRAMMING		
DATE	Monday 18 th August 2021	TIME ALLOWED: 3 HOURS 02:00 PM - 05:00PM	

INSTRUCTIONS TO STUDENT

1. This paper consists of **TWO (2)** sections **A** and **B** which carry 24 marks and 36 marks respectively.
2. Answer **ALL** questions from section **A** and any **THREE (3)** questions from section **B**.
3. Cellular **PHONES** and any other **UNAUTHORIZED** materials are **NOT allowed** in the examination room.
4. Don't write anything in this paper.
5. **START EACH QUESTION IN A NEW PAGE.**
6. This examination consists of **8** printed pages, including the cover page

SECTION A (24 marks)
Answer ALL questions

[10 marks]

1. Please answer the following true/false questions. give a brief explanation of answer.

- i. Class and object mean the same thing.
- ii. An interface can be used as the type of a variable.
- iii. A method signature includes information about the number and types of parameters that a method has.
- iv. To be notified when a user clicks on a button, the program must add a listener to the button.
- v. A subclass can add behavior that is not present in the superclass.
- vi. If FileNotFoundException occurs when a program is executed, it means that the program contains a bug.
- vii. Instance variables should always be declared to be public.
- viii. 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".)
- ix. Calling nextInt() on a Scanner object two times in a row will always return the same value.
- x. An exception can be thrown in one method and handled in the method that called it.

2. [8 marks]

```
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: " + x);  
    }  
    public static void printCount(){  
        System.out.println("Value of count: " + count);  
    }  
}
```

```
public class MyClassDemo{  
    public static void main(String[] args){  
        MyClass myObject1 = new MyClass(5);  
        MyClass myObject2 = new MyClass(7);  
    }  
}
```

- a. myObject1.printX();
- b. myObject1.incrementCount();
- c. MyClass.incrementCount();
- d. myObject1.printCount();
- e. myObject2.printCount();
- f. myObject2.printX();
- g. myObject1.setX(14);
- h. myObject1.incrementCount();
- i. myObject1.printX();
- j. myObject1.printCount();
- k. myObject2.printCount();

3. What is the output of the following:

[6 marks]

a.
 int a = 6, b = a/4, c = a % 4;
 float d = a/4.0;
 System.out.println("a = " + a + b = " + c + "d = " + d);

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

 c. 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");

SECTION B (36 marks)**Answer any three Questions, Each question carries 12 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.
 - 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 artistComposer 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 totalLength, 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
- b. Write a static method called swap that swaps two elements of an int array. The method should accept three input parameters:
- o an int representing the index of one element being swapped
 - o an int representing the index of the other element being swapped
 - o 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 [4, 3, 2, 1].
- 5.
- 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 variables 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 [1 mark]

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. [2 marks]

6 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.

7 a. Consider the following interface and class declarations. For each part of the question that follows, indicate if the statement is valid or not. Consider a statement to be valid if it would compile and execute without throwing an exception, given just the information shown below. If it is not valid, briefly explain why not. You can assume that the interfaces and classes shown all compile and work correctly. ... means that part of the definition is not shown because it is not relevant to the question.

```
public interface Shape {  
    ...  
}  
  
public class Polygon implements Shape {  
    /** Creates a new polygon. */  
    public Polygon() {...}  
    ...  
}  
  
public class Rectangle implements Shape {  
    /** Creates a new Rectangle with the given width and height. */  
    public Rectangle (int width, int height) {...}  
    ...  
}
```

- i. Shape s = new Shape();
- ii. Shape s = new Rectangle(3, 4);
- iii. Polygon p = new Rectangle (3, 4);
- iv. Polygon p = new Polygon();
- v. Polygon p = new Shape();

b. A Rectangle can be implemented as a class with two instance variables: length and width:

```
public class Rectangle {  
    private int length;  
    private int width;  
}
```

- i. 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.
- ii. Create a constructor which has default values set to 1.
- iii. Create a second constructor which pass length and width and throws NegativeValueException if length or width is less than 0.
- iv. Provide a set function which set both length and width and throws an exception if length and width is less than 0.
- v. Provide getArea function which returns area of rectangle an getPerimeter functions which returns perimeter of rectangle.

8. Do the following for the subclass Dog:
- i. Define the default constructor so it sets hasTail to true, numLegs to 4, and mass to 10.0, as in the parent class.
- ii. Create a setter for mass that throws an Exception if the input parameter is less than 2 or greater than 350.
- iii. 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.
- [12 marks]