System Analysis & Design
Quiz 2

11/9

Name: 許家瑋 Jobe ID: B10302002

1. Please describe the differences for the following statements and their applic

6%

```
public class Duck implements Animal
public class Duck extends Animal
```

2. Please use examples to describe poor inheritance by using IS-A test and conformance test. 10%
3. Create a CRC card for each of the following classes: 10%
   Movie (title, producer, length, director, genre)
   Ticket (price, adult or child, showtime, movie)
   Patron (name, adult or child, age)

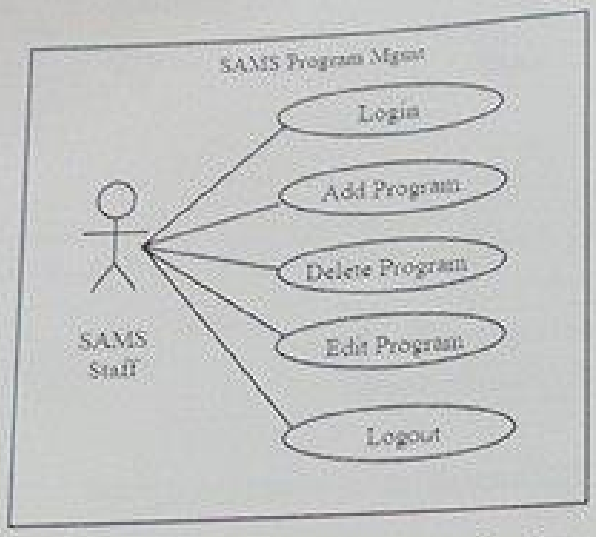
Front:

Class Name:	**ID:**	**Type:**
Description:		**Associated Use**
Responsibilities		**Collabo**

Back:

Attributes:
Relationships: **Generalization (a-kind-of):** **Aggregation (has-parts):** **Other Associations:**

4. What is the problem of the following use case diagram? 5%



5. An ATM system includes the main functions of withdraw cash, deposit funds, check balance, transfer funds, perform routine maintenance, and fill ATM with cash. Please draw a use case diagram 3% and describe the normal flow of events, subflows, alternate/Exception flows for the "withdraw cash" function 15%.

6. Draw an **activity diagram** for a health center system: Whenever new patients are seen for the first time, they complete a patient information form that asks their name, address, phone number and brief medical history, which are stored in the patient information file/ When a patient calls to schedule a new appointment or change an existing appointment, the receptionist checks the appointment file for an available time. Once a good time is found for the patient, the appointment is scheduled. If the patient is a new patient, an incomplete entry is made in the patient file; the full information will be collected when they arrive for their appointment 10%.

7. Use examples to explain the difference between an activity and an action. 5%

8. Given the following code, what will be the output? 5% What is the implication of this example? 5%

```
public abstract class Quadrilateral {
    protected int m_width;
    protected int m_height;
    public void setWidth(int width) {
        m_width = width;
    }
    public void setHeight(int height) {
        m_height = height;
    }
    public int getM_width() {
```

```
        return m_width;
    }
    public void setM_width(int m_width) {
        this.m_width = m_width;
    }

    public int getM_height() {
        return m_height;
    }
    public void setM_height(int m_height) {
        this.m_height = m_height;
    }
    public int getArea() {
        return m_width * m_height;
    }}

public class Rectangle extends Quadrilateral{
}

public class Square extends Quadrilateral{
    public void setWidth(int width) {
        m_width = width;
        m_height = width;
    }
    public void setHeight(int height) {
        m_height = height;
        m_width = height;
    } }

public class Test {
    private static Quadrilateral getQuadrilateral() {
        return new Square();
    }
        public static void main(String[] args) {
    {
        Quadrilateral q = Test.getQuadrilateral();
        q.setWidth(5);
        q.setHeight(10);
```

```
        System.out.println(q.getArea());
    } } }
```

9. Given the following code, please indicate the relationship for class Person and Job 2% and explain the reason 5%

```
public class Person {
    private Job job;

    public Person(){
        this.job=new Job();
        job.setSalary(1000L);
    }
    public long getSalary() {
        return job.getSalary();
    }
}

public class Job {
    private String role;
    private long salary;
    private int id;

    public String getRole() {
        return role;
    }
    public void setRole(String role) {
        this.role = role;
    }
    public long getSalary() {
        return salary;
    }
    public void setSalary(long salary) {
        this.salary = salary;
}
public int getId() {
    return id;
```

```
    }
    public void setId(int id) {
        this.id = id;
    } }
```

10. Given the code below, please do the following: 1) draw the class diagram for class Purchase and PrintDetail 5% and explain their relationship 2%; 2) please indicate why overloading is used in this example 5%.

```
public class Purchase {
    private int billId;
    private float billAmount;
    public Purchase(int billId, float billAmount){
        this.billId=billId;
        this.billAmount=billAmount;
    }
    public void calculateBill(String modeOfPayment, int processingCharge){
        //logic for bill calculation
    }
    public void displayBill(){
        PrintDetails printObj=new PrintDetails();
        printObj.printHeader('*');
        printObj.printHeader('-',70);
        printObj.printHeader(" Cloud Retail Store Bill");
        System.out.println("");
        System.out.println("Bill Id :"+billId);
        System.out.println("Final bill amount to be "+"paid :Rs.
"+billAmount);
        System.out.println("");
        printObj.printHeader('-',70);
        printObj.printHeader("Thank you!!");
    } }

public class PrintDetails {
    public void printHeader(char c){
        for(int counter=0; counter<70; counter++){
            System.out.print(c);
```

```
    } }
    public void printHeader(char c, int no){
        for(int counter=0; counter<no; counter++){
            System.out.print(c);
        } }
    public void printHeader(String s){
        System.out.println(s);
    } }

public class Client {
    public static void main(String args[]) {
        String modeOfPayment=args[0];
        int processingCharge=Integer.parseInt(args[1]);
        Purchase purObj=new Purchase(1001, 500f);
        purObj.calculateBill(modeOfPayment, processingCharge);
        purObj.displayBill();
    } }
```

11. Given the code below, please indicate the relationship for class Customer and Address 2% and explain the reason 5%.

```
public class Customer {
    private int customerId;
    private Address addressLine;
    public Customer(int customerId, Address addressLine) {
        this.customerId=customerId;
        this.addressLine=addressLine;
    }
    public int getCustomerId(){
        return customerId;
    }
    Public Address getAddressLine(){
        return addressLine;
    } }
public class Address {
    private int doorNo;
```

```
    private String locality;
    public Address(int doorNo, String locality){
        this.doorNo=doorNo;
        this.locality=locality;
    }
    public int getDoorNo(){
        return doorNo;
    }
    public String getLocality(){
        return locality;
    } }

public class Main {

    public static void main(String[] args) {
        String locality=new String("Cloud Road, Happy Town, Taiwan,
R.O.C.");
        Address add=new Address(123, locality);
        Customer david=new Customer(1001, add);
        System.out.println("Customer Id: "+david.getCustomerId());
        System.out.println("Customer Address: ");
        System.out.println("Door No:
"+david.getAddressLine().getDoorNo());
        System.out.print("Locality:
"+david.getAddressLine().getLocality());
    } }
```