

22nd SCSE – Past Year Paper Solution (2021 – 2022 Semester 1)
CE/CZ 2002 – Object-Oriented Design & Programming

1 (a) (2)

There is only one 'Square' object created.

's2 = s1' only pass the reference to the Square object to s2. Both s1 and s2 are referring to the same object

(b) (5)

Line (ii) cause an error due to Java doesn't support print() function with no signatures in it.

Line (iii) cause an error due to conflicting characteristics between 'private' and 'abstract'. You cannot access 'private' outside the current class, not even from the child classes of it. You must override 'abstract' inside the child class, or else code would not work.

(c) (2)

The keyword here is compilation error. aa is ClassX contains the reference to a ClassY object. If aa tries to run the doC() function, it will resulting in compilation error.

(d) (5).

If access within same class, all access levels are fine.

If access within same package, public(+), protected(#), and default() are fine

If access with subclass in different package, public(+), and protected(#) are fine

Else, public(+) will do

(e) (2)

This is an example of Method Overloading, where talk() function ends up with different result with different parameters assigned to it.

(f) (3)

Student is a subclass of Person, so Upcasting is allowed

Object p with class Person is able to run the function talk with int and String passed it.

So, resulting in runtime OK

2 (a) Assumption : Sum of percentage of courseComponent in a course is 100

```
public class CourseComponent{
    private String componentName;
    private double percentage;
    private double score;
    public CourseComponent(String name, double percent, double score){
        this.componentName = name;
        this.percentage = percent;
        this.score = score;
    }
    public double getPercentage(){ return this.percentage;}
    public double getScore(){ return this.score;}
```

22nd SCSE – Past Year Paper Solution (2021 – 2022 Semester 1)
CE/CZ 2002 – Object-Oriented Design & Programming

```
}
```

```
(b) public abstract class Student{  
    private String name;  
    public Student(String name){  
        this.name = name;  
    }  
    public abstract double calOverall();  
}
```

```
(c) public class Undergraduate extends Student{  
    private CourseComponent assignment;  
    private CourseComponent finalExam;  
    public Undergraduate(String name, CourseComponent assignment,  
        CourseComponent finalExam){  
        super(name);  
        this.assignment = assignment;  
        this.finalExam = finalExam;  
    }  
    public double calOverall(){  
        double sum = 0;  
        sum += assignment.getPercentage()*assignment.getScore();  
        sum += finalExam.getPercentage()*finalExam.getScore();  
        return sum/100.0;  
    }  
}  
  
public class Graduate extends Student{  
    private CourseComponent assignment;  
    private CourseComponent finalExam;  
    private CourseComponent researchReport;  
    public Graduate(String name, CourseComponent assignment, CourseComponent  
        finalExam, CourseComponent researchReport){  
        super(name);  
        this.assignment = assignment;  
        this.finalExam = finalExam;  
        this.researchReport = researchReport;  
    }  
    public double calOverall(){  
        double sum = 0;  
        sum += assignment.getPercentage()*assignment.getScore();  
        sum += finalExam.getPercentage()*finalExam.getScore();  
        sum += researchReport.getPercentage()*researchReport.getScore();  
        return sum/100.0;  
    }  
}
```

22nd SCSE – Past Year Paper Solution (2021 – 2022 Semester 1)
CE/CZ 2002 – Object-Oriented Design & Programming

}

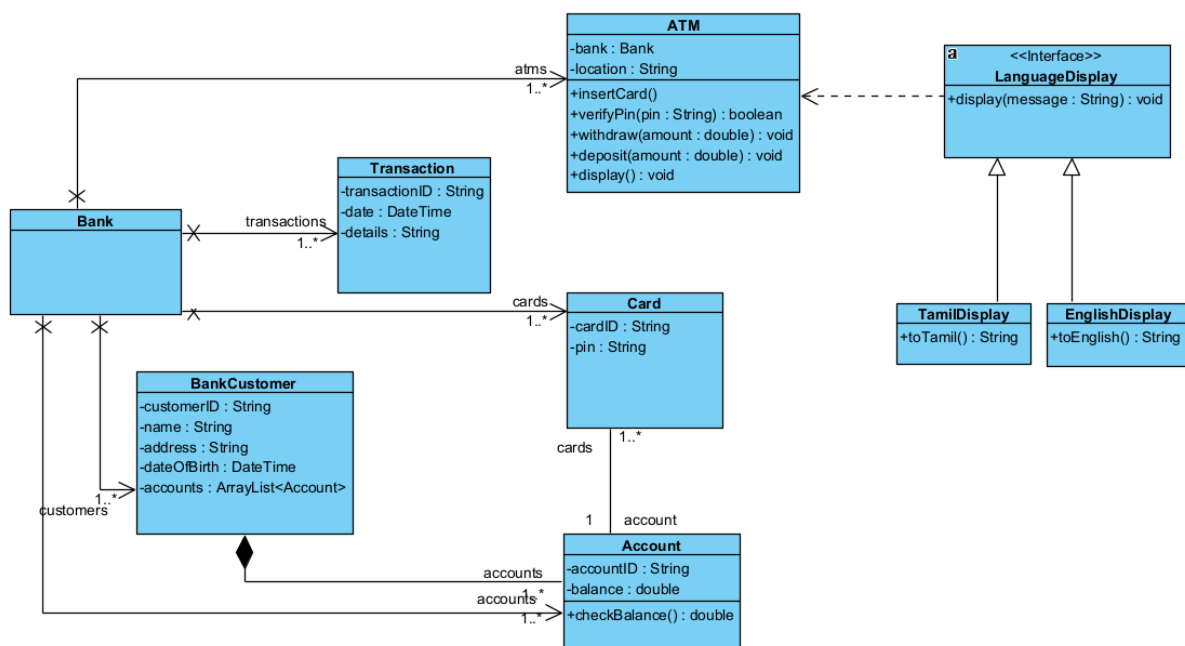
```
(d) public class StudentApp{
    public static void printOverall(Student student){
        System.out.println("Average\t: " + student.calOverall());
    }
    public static void main(String []args){
        CourseComponent assignment = new CourseComponent("assignment1",50,90);
        CourseComponent finalExam = new CourseComponent("finalExa1",50,80);

        CourseComponent assignment2 = new CourseComponent("assignment2",30,100);
        CourseComponent finalExam2 = new CourseComponent("finalExam2",40,100);
        CourseComponent research = new CourseComponent("research1",30,100);

        Undergraduate student1 = new Undergraduate("Alex",assignment,finalExam);
        printOverall(student1);

        Graduate student2 = new Graduate("Bendy", assignment2, finalExam2,
        research);
        printOverall(student2);
    }
}
```

3 (a)



(b) Open-Closed Principle

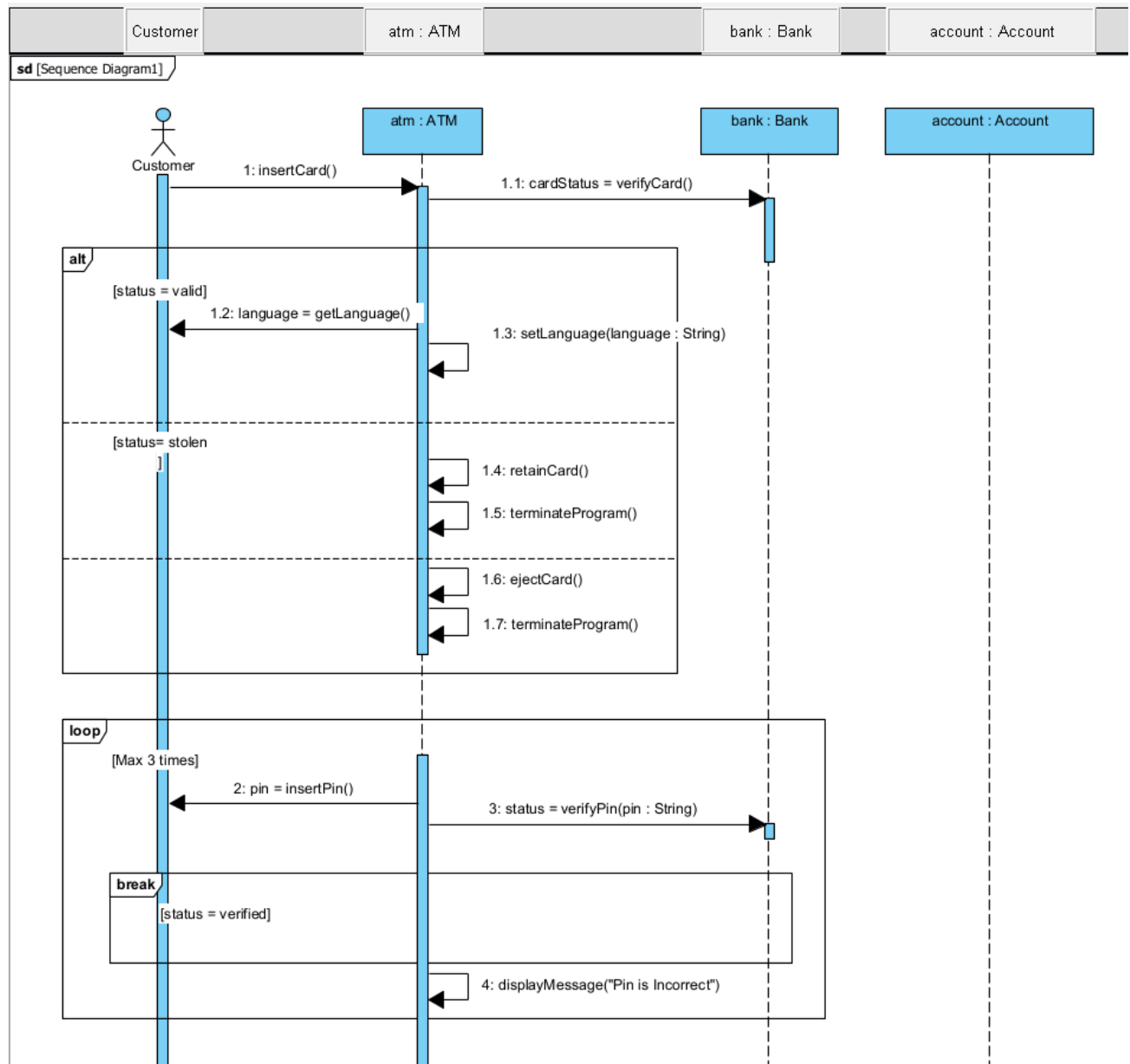
22nd SCSE – Past Year Paper Solution (2021 – 2022 Semester 1) CE/CZ 2002 – Object-Oriented Design & Programming

A module is open for extension but closed for modification. This allows us to add new functionality without the need to modify the original source code. For example, LanguageDisplay will display message according to the language chosen. By extending from LanguageDisplay, we can create other language display with different features but with basic functionality.

Interface Segregation Principle:

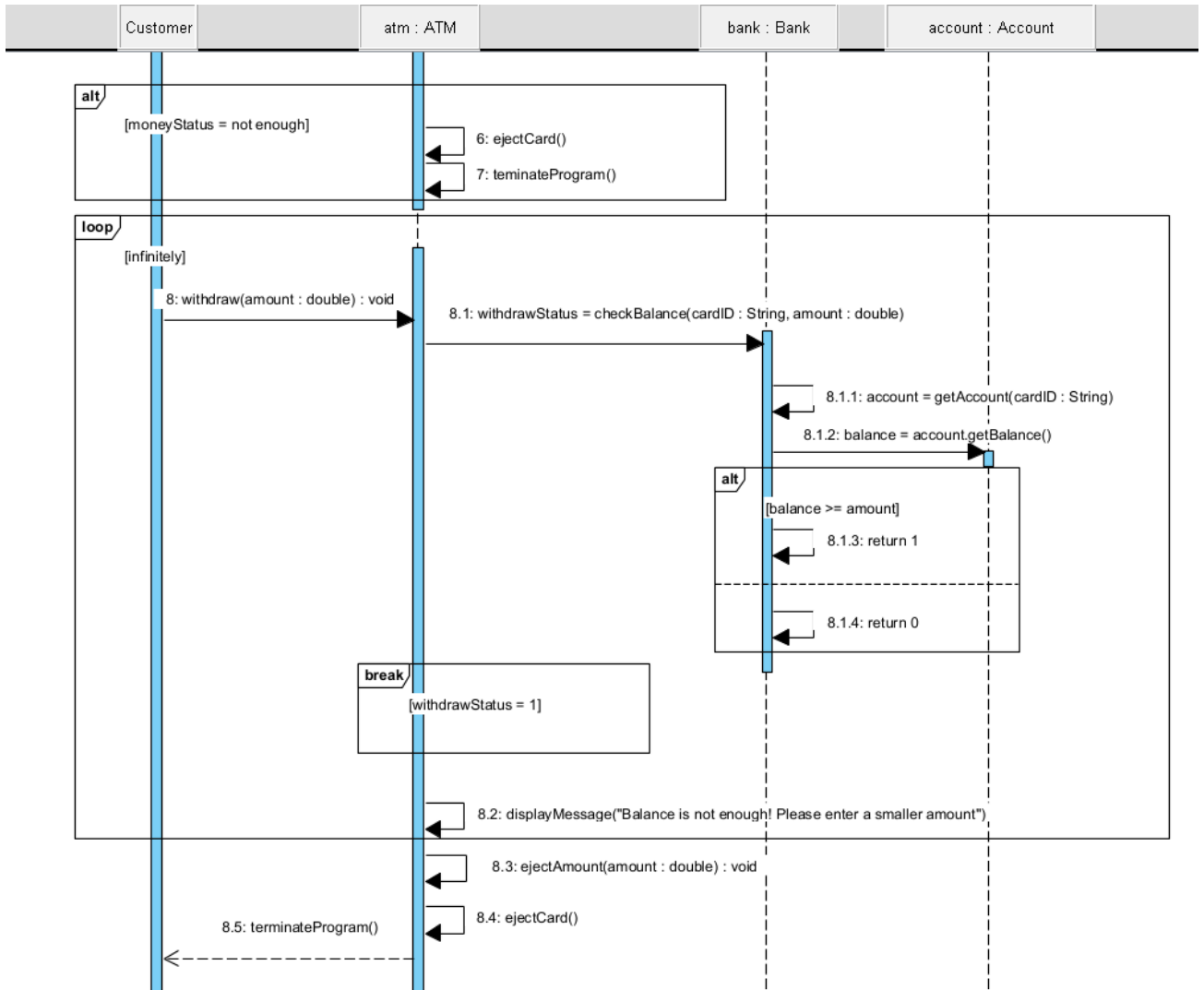
This is done by creating a LanguageDisplay interface so the ATM only relies on the interfaces but not the subclasses. If done correctly, modifications of LanguageDisplay will require less effort when any modification of ATM are done or changes to the way of displaying.

4 (a) Step 1 – 3 :



Step 4 – 6 :

22nd SCSE – Past Year Paper Solution (2021 – 2022 Semester 1)
CE/CZ 2002 – Object-Oriented Design & Programming



```

(b) #include <iostream>
#include <string>

class iLivingThing{
public:
    virtual void grow() = 0;
};

class Animal : public iLivingThing{
private:
    int age;
public:
    Animal(){};
    Animal(int age): age(age){};
    int getAge(){ return age;}
}
    
```

22nd SCSE – Past Year Paper Solution (2021 – 2022 Semester 1)
CE/CZ 2002 – Object-Oriented Design & Programming

```
void grow(){ std::cout << "Eat well to grow" << std::endl; }
virtual void speak() {std::cout << "Sound" << std::endl; }
};

class Cat : public Animal{
private :
    std::string name;
public :
    Cat(int age, std::string name):Animal(age), name(name){};
    void speak() { std::cout << "Meow" << std::endl; }
};

class Dog : public Animal{
public :
    Dog():Animal(){};
};

int main()
{
    Animal* a1 = new Cat(12, "Snow");
    std::cout << a1->getAge() << std::endl;
    a1->speak();
    a1->grow();
    Animal* a2 = new Dog();
    std::cout << a2->getAge() << std::endl;
    a2->speak();
    a2->grow();
}
```

Solver: Tan Caken (C200243@e.ntu.edu.sg)