(Attempt 2 questions out of 3)

Question 1.

(a) Discuss the object technology concept of **polymorphism** and its usefulness in software engineering using example code and diagrams where appropriate.

[**8 marks**]

(b) Consider a class **Student** which has features of name (a String) and number (an int). Write a constructor which takes the necessary parameters and sets the features accordingly. Write a **default** constructor which takes no parameters and sets the features to reasonable default values.

[**4 marks**]

(c) Consider the following **Abstract Data Type** specification **Stack** for an unbounded Stack of integers.

```
Stack(); //Constructor for the empty stack
int top(); //The integer at the front of the stack
void push(int i); //Push integer i onto the stack
void pop(); //Remove the first integer from the stack
Boolean empty(); //Is the stack empty?
```

Write a class **Stack** which implements this ADT. You should use an **ArrayList** to contain the data and will probably need to use the ArrayList methods **add** and **removeElementAt(int i)**, as well as the default constructor. Illegal operations should print a warning message to the screen and leave the stack unchanged. Full marks will be given for correct code including return values and visibility modifiers.

[**9 marks**]

(d) What is meant by the following modifiers in Java. Your answer should give reasons and examples for their use.

 – final

 – static

[**4 marks**]

Question 2.

(a) Discuss the meaning of, and the difference between, **static** and **dynamic** typing.

[**8 marks**]

(b) Explain the meaning of the Object Orientated concept of **encapsulation** of data. Contrast encapsulation with the procedural programming view of data and explain how encapsulation helps with software engineering.

[**8 marks**]

(c) Explain, with the aid of suitable examples, the difference between method *overloading* and method *overriding*.

[**5 marks**]

(d) Explain why a constructor cannot be declared as

– final
– abstract

[**4 marks**]

Question 3.

(a) Explain what is meant in Java by an **interface**. Why would an interface be used?

[**8 marks**]

(b) Explain where a method is visible if it is declared as

– public
– private
– protected

[**6 marks**]

(c) Consider the following two classes :-

```
public class Account{
    private String name;
    private int number;
    //other features

    public Account(String name, int number){
        this.name = name;
        this.number = number;
    }

    //other methods
}

public class InterestAccount extends Account{
    //various features
    private int interestRate;

    public InterestAccount(int rate, String name, int number){
        this.name = name;
        this.number = number;
        interestRate = rate;
    }
}
```

Why would this code fail to compile? Alter the code in either class to ensure that compilation will succeed.

[6 marks]

(d) Explain how object oriented design lends itself to the concepts of **Design By Contract**. Explain how **require** and **ensure** clauses can be simulated in languages which do not have them.

[5 marks]