Abstraction:

Its when something complex is represented in a simple manner that allows users to understand them easily. In JAVA, things such as objects and classes represent the details of complex data and code

that exist within them, functionality of these objects and classes is the part that users need to

understand. During implementaion the user will only see what the object does rather than how it does it.

Inheritance:

Its when attributes(properties) that exist from a class can share and adapt with a new class.

The class that is inherting the attributes are referred to as subclass or a child class and

the original class is referred to as a parent. The word extends is often used to define

a child class is inherting attributes or properties from the parent class.

Example:

class Bird {

}

class Pigeon extends Bird {

}

Polymorphism:

In JAVA its when multiple child classes or sub classes inherits similar functionality from thier

parent class. It allows these sub classes to act on thier own unique behaviour(method) whilst still maintinining functionality inhereted from thier parent class.

Example:

class Mammal {

void chase() {

System.out.println(“Can Chase….”);

}

}

class Leopard extends Mammal {

void chase() {

System.out.println(“Chase Fast…”);

}

public static void main(String arg[]) {

Mammal m = new Leopard();

m.chase();

}

}

Encapsulation:

Its when variables and methods can be from a class can be kept hidden and prevent access to

other classes. Encapsulation can be done by declaring the variables of a class as private or

providing public getter and setter methods to access the values of these variables.

Example:

public class Human {

private String person;

public String getPerson() {

return person;

}

public void askPerson(String person) {

this.person = person

}

}

class CheckHuman {

public static void main(String[] args) {

person p = new person();

p.askPerson(“Rick Deckard”);

System.out.println(s.getPerson());

}

}