Name: Abdul Hamid Piash
ID: IT22021

(1) Think of two things,

(i) A Button - it can be clicked.

(ii) A Remote Control - it can be clicked and it has a battery.

Here we can use interface and abstract class;

Code:

```
interface Switch{
    void click();
}
abstract class BatterySwitch implements Switch{
    void checkBattery(){
        System.out.println("Battery is full.");
    }
}
class SimpleSwitch implements Switch{
    public void click(){
        System.out.println("Light turned ON!");
    }
}
```

```
class RemoteSwitch extends BatterySwitch{
    public void click(){
        checkBattery();
        System.out.println(" Remote light turned ON
    }
}

public class Abstraction{
    public static void main(String[] args){
        Switch S1 = new SimpleSwitch();
        S1.click();
        Switch S2 = new S.RemoteSwitch();
        S2.click();
    }
}
```

Here SimpleSwitch inheritance from interface and RemoteSwitch inheritance from an abstract class BatterySwitch.

(2) Calling a method from an interface can be slightly slower than from an abstract class because the JVM uses a different lookup mechanism. Interfaces use an interface table which abstract classes use a virtual table. The vartual table is usually faster to access.

Ex:

Code:

```
interface PlayableCharecter{
    void play();
}
abstruct class Fighter{
    void attack(){
        System.out.println("Attack with sword!");
    }
}
class Hero implement PlayableCharecter {
    public void play(){
        System.out.println("Hero is playing the game");
    }
}
```

```
class Warrior extends Fighter{

}

public class Abstraction{
    public static void main(String[] args){
        playableCharecter h = new Hero();
        h.play();

        Warrior
        Warrion w = new Warrion();
        w.attack();
    }
}
```

Here Hero() class is slower than Warrior class.

(3) The difference between Abstract class and Interface is:

Feature	Abstract class	Interface
Method type	Abstract and Concrete method	Abstract (default static from java.
Abstraction	Here can be abstruction classes or normal classes or both.	Strictly have abstract classes
Object Creation	Can't be created	Can't be created.
Multiple Inheritance	Not supported	Supported.
Access Modifiers	Can be use public, protected etc.	All methods are public by default.
Inheritance keyword	Extends	Implements.