1+) When to use Interface and when to abstract class. Develop a story and write codes to explain

Ans: Lets developing an online payment system that handles multiple payment methods such as credit card, pay pal and upi.

*All payment method must support pay (amount). this is common action. but how its implemented varies.

* Some payment methods can optionally support refunds but not at all thats an optional behaviour.

Design approch:

(i) Use and interface Refundable for payment that can be refunded. Its not application to all pay

(ii) we an abstract application to all p

code

```
Interface refundable {
    void refund (double amount)
}
abstract class creditpayment {
    String cardNumber;
    credit card payment (String cardNumber) {
        this.cardNumber = cardNumber;
    }
    void validate card () {
        System.out.println ("validating card " + card number)
    }
    abstract void pay (double amount);
}
class visacardpayment (String card number) {
    super (cardnumber);
}
public void pay (double amount) {
    validate card ();
    System.out.println ("paid " + amount + " using")
}
class paypalpayment implements refundable {
    public void pay (double amount) {
        System.out.println ("paid $" + amount");
    }
    public void refund (double amount) {
        System.out.println ("Refunded $" + amount + "via
                                                    paypal");
    }
}
```

12) Is it true that involving method in an interface is slower than abstract class method?

Ans: No, It is no longer meaningfully true in modern java (post Java 7+). Interface method calls were slightly slower in early JVMs because of dynamic dispatch but in modern JVM this is different. This difference is negligible.

13) Make a table to summarize the difference between abstract class and interface.

Interface	Abstract class
1) Define a contract for behaviour	1) Provid class with sh code
2) Can implement multiple interface	2) Only one abstract can be extended
3) Can not have construction	3) Can have construc
4) Only public static final (constant)	4) Can have instance variable