Q1.

```
abstract class bank {
    abstract void getBalance();
}
class BankA extends bank {
    public void getBalance(){
        System.out.println("$100 deposited");
    }
}
class BankB extends bank {
    public void getBalance(){
        System.out.println("$150 deposited");
    }
}
class BankC extends bank {
    public void getBalance(){
        System.out.println("$200 deposited");
    }
}
public class q1 {
    public static void main(String[] args) {
        bank obj = new BankA();
        obj.getBalance();
        bank obj1 = new BankB();
        obj1.getBalance();
        bank obj2 = new BankC();
        obj2.getBalance();
    }
}
```

Q2.

```
abstract class Marks {
    abstract float getPercentage();
}

class A extends Marks {
    float sub1, sub2, sub3;
    A(float sub1, float sub2, float sub3) {
        this.sub1 = sub1;
        this.sub2 = sub2;
        this.sub3 = sub3;
    }
    float getPercentage() {
        float total_marks = sub1 + sub2 + sub3;
        float percentage = (total_marks / 300) * 100;
        return percentage;
    }
}
```

```
class B extends Marks {
    float sub1, sub2, sub3, sub4;
    B(float sub1, float sub2, float sub3, float sub4) {
        this.sub1 = sub1;
        this.sub2 = sub2;
        this.sub3 = sub3;
        this.sub4 = sub4;
    }
    float getPercentage() {
        float total_marks = sub1 + sub2 + sub3 + sub4;
        float percentage = (total_marks / 400) * 100;
        return percentage;
    }
}

public class q2 {
    public static void main(String[] args) {
        A student_a = new A(80, 85, 90);
        B student_b = new B(75, 80, 85, 90);

        System.out.println("Percentage of marks for Student A: " +
student_a.getPercentage());
        System.out.println("Percentage of marks for Student B: " +
student_b.getPercentage());
    }
}
```

Q3.

```
abstract class character{
    public int health;
    public int attackpower;
    public String name;
    abstract void specialability();
    public void attack(character obj){
        this.health = attackpower;
    }
    public void defend(character obj){
        System.out.println("defend");
    }

    public character(int health, int attackpower, String name) {
        this.health = health;
        this.attackpower = attackpower;
        this.name = name;
    }
}
```

```
class Warrior extends character{
    public Warrior(int health, int attackpower, String name) {
        super(health, attackpower, name);
    }

    public void specialability(){
        attackpower = attackpower*2;
    }
}
class mage extends character{
    public mage(int health, int attackpower, String name) {
        super(health, attackpower, name);
    }

    public void specialability(){
        attackpower = attackpower*20;
    }
}
class archer extends character{
    public archer(int health, int attackpower, String name) {
        super(health, attackpower, name);
    }

    public void specialability(){
        attackpower = attackpower*15;
    }
}
public class q3 {
    public static void main(String[] args) {
        character warrior = new Warrior(100,20,"Fatima");
        character mage = new mage(50, 28,"Laiba");
        character archer = new archer(89, 30, "Noor");
        warrior.attack(mage) ;
        System.out.println(mage.health);
        mage.defend(warrior);
        warrior.specialability();
        System.out.println(warrior.attackpower);
        archer.specialability();
        System.out.println(archer.attackpower);
        archer.attack(warrior);
        System.out.println(warrior.health);
        warrior.defend(archer);


    }
}
```

Q4.

```
abstract class Animals {
    public abstract void cats();
    public abstract void dogs();
}

class Cats extends Animals {
    public void cats() {
        System.out.println("Cats meow");
    }
    public void dogs() {
        // Do nothing, as Cats don't bark.
    }
}

class Dogs extends Animals {
    public void cats() {
        // Do nothing, as Dogs don't meow.
    }
    public void dogs() {
        System.out.println("Dogs bark");
    }
}

public class q4 {
    public static void main(String[] args) {
        Animals cat = new Cats();
        cat.cats(); // Output: "Cats meow"

        Animals dog = new Dogs();
        dog.dogs(); // Output: "Dogs bark"
    }
}
```

Q5.

```
import java.util.Scanner;
interface MessageService{
    void send(String message,String recipient);
    void recieve();
}
class Whatsapp implements MessageService{
    String message;
    String recipient;
    @Override
```

```
    public void send(String message, String recipient)
    {
        this.message=message;
        this.recipient=recipient;
        System.out.println("Message sent");
    }
    @Override
    public void recieve() {
        System.out.println("Message sent to "+recipient);
        System.out.println(message);
    }
}
public class q5 {
    public static void main(String[] args) {
        Scanner A =new Scanner(System.in);
        System.out.println("Enter message: ");
        String message= A.next();
        System.out.println("Enter recipient: ");
        String recipient= A.next();
        Whatsapp w=new Whatsapp();
        w.send(message,recipient);
        w.recieve();
    }
}
```

Q6.

```
import java.util.Scanner;
 interface BankAccount {
    void deposit(double amount);
    void withdraw(double amount);
    double getBalance();
}
 class SavingsAccount implements BankAccount {

    private double balance;

    @Override
    public void deposit(double amount) {
        balance += amount;
    }

    @Override
    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
        } else {
            System.out.println("Insufficient balance!");
        }
    }

    @Override
```

```
    public double getBalance() {
        return balance;
    }
}


public class q6 {

    public static void main(String[] args) {

        SavingsAccount savingsAccount = new SavingsAccount();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("Menu:");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Get balance");
            System.out.println("4. Exit");

            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");
                    double depositAmount = scanner.nextDouble();
                    savingsAccount.deposit(depositAmount);
                    System.out.println("Amount deposited successfully!");
                    break;
                case 2:
                    System.out.print("Enter amount to withdraw: ");
                    double withdrawAmount = scanner.nextDouble();
                    savingsAccount.withdraw(withdrawAmount);
                    System.out.println("Amount withdrawn successfully!");
                    break;
                case 3:
                    System.out.println("Current balance: " +
savingsAccount.getBalance());
                    break;
                case 4:
                    System.out.println("Thank you for using our
application!");
                    System.exit(0);
                default:
                    System.out.println("Invalid choice!");
            }
        }
    }
}
```