Main:

```
package com.torryharris.bankpack;

import com.torryharris.accountpack.Account;

public class Main {

    public static void main(String[] args) {
   // write your code here
        Account account1 = new Account("savings",5000);
        System.out.println(account1);
        System.out.println("amount deposited: "+account1.deposit(6000));
        System.out.println("current account state: \n"+account1);
        System.out.println("amount withdrawn: "+account1.withdraw(1000));
        System.out.println("current account state: \n"+account1);
        System.out.println("---------------------------------");
        Account account2 = new Account("Recurring",6000);
        System.out.println(account2);
        System.out.println("amount deposited: "+account2.deposit(6000));
        System.out.println("current account state: \n"+account2);
        System.out.println("amount withdrawn: "+account2.withdraw(1000));
        System.out.println("current account state: \n"+account2);
    }
}
```

account class:

```
package com.torryharris.accountpack;

import java.util.Random;

public class Account {
    private static long count;
    private final long accno;
    private String acctype;
    private double balance;

    static {
        count=0;
    }
    public static long getCount()
    {
        Random rand = new Random();
        count = rand.nextInt(1000)+3000;
        return count;
    }

    public Account(String acctype, double balance) {
        this.accno = getCount();
        this.acctype = acctype;
        this.balance = balance;
    }
    public double deposit(double amount)
```

```
    {
        balance+=amount;
        return amount;
    }
    public double withdraw(double amount)
    {
        balance-=amount;
        return amount;
    }
    public double getBalance()
    {
        return balance;
    }

    @Override
    public String toString() {
        return "Account{" +
                "accno=" + accno +
                ", acctype='" + acctype + '\'' +
                ", balance=" + balance +
                '}';
    }
}
```

```
Output:

Account{accno=3012, acctype='savings', balance=5000.0}

amount deposited: 6000.0

current account state:

Account{accno=3012, acctype='savings', balance=11000.0}

amount withdrawn: 1000.0

current account state:

Account{accno=3012, acctype='savings', balance=10000.0}

--------------------------------

Account{accno=3249, acctype='Recurring', balance=6000.0}

amount deposited: 6000.0

current account state:

Account{accno=3249, acctype='Recurring', balance=12000.0}

amount withdrawn: 1000.0

current account state:

Account{accno=3249, acctype='Recurring', balance=11000.0}

================================================================================
```

Main:

```
package com.torryharris.mainpack;

import com.torryharris.emppack.employee;
import com.torryharris.emppack.manager;
import com.torryharris.emppack.programmer;
import com.torryharris.taxpack.tax;

public class Main {

    public static void main(String[] args) {
   // write your code here
       /* employee e1 = new employee(100,"varun",10000);
        System.out.println(e1.getDetails());

        manager m1 = new manager(200,"suhas",40000,"HR",5);
        System.out.println(m1.getDetails());

        programmer p1 = new programmer(300,"ajay",30000,3,"java");
        System.out.println(p1.getDetailes());

        employee e2 = new manager(400,"abhi",50000,"IT",6);
        System.out.println(e2.getDetails());

        manager m2 = (manager) e2;
        System.out.println("works for: "+m2.getDetails()+" department");*/

        employee[] emparr = new employee[4];
        emparr[0] = new employee(100,"Joe",20000);
        emparr[1] = new manager(200,"suhas",30000,"IT",5);
        emparr[2] = new programmer(300,"ajay",25000,3,"java");
        emparr[3] = new programmer(301,"varun",25000,4,"python");

        System.out.println("printing employee details");
        for(employee e:emparr)
        {
            System.out.println(e.getDetails()+" tax paid :"+ tax.calc_tax(e));
        }


    }
}
```

Employee class(base):

```
package com.torryharris.emppack;

public class employee {
```

```
    protected int empid;
    protected String empname;
    protected int sal;


    public employee(int empid, String empname, int sal) {
        this.empid = empid;
        this.empname = empname;
        this.sal = sal;
    }
    public String getDetails()
    {
        return(empid+"  "+empname+"  "+sal);
    }
    public int getSal(){
        return sal;
    }

    @Override
    public String toString() {
        return "employee{" +
                "empid=" + empid +
                ", empname='" + empname + '\'' +
                ", sal=" + sal +
                '}';
    }

}
```

Manager class(sub):

```
package com.torryharris.emppack;

public class manager extends employee{
    private String depname;
    private int empcount;

    public manager(int empid, String empname, int sal,String depname,int empcount) {
        super(empid, empname, sal);
        this.depname=depname;
        this.empcount=empcount;
    }
    public String getDetails()
    {
        return(super.getDetails()+"  "+depname+"  "+empcount);
    }
    public String getDepname(){
        return depname;
    }
    public int getEmpcount(){
        return empcount;
    }
```

```
}
```

Programmer class(sub):

```
package com.torryharris.emppack;

public class programmer extends employee{
    private int noofproj;
    private String skillset;

    public programmer(int empid, String empname, int sal,int noofproj,String skillset) {
        super(empid, empname, sal);
        this.noofproj=noofproj;
        this.skillset=skillset;

    }

    @Override
    public String getDetails() {
        return super.getDetails()+"   "+noofproj+"   "+skillset;
    }
}
```

Tax class(sub):

```
package com.torryharris.taxpack;

import com.torryharris.emppack.employee;
import com.torryharris.emppack.manager;
import com.torryharris.emppack.programmer;

public class tax {
    public static double calc_tax(employee e)
    {
        if (e instanceof manager)
            return (e.getSal()*0.2);
        else
            if(e instanceof programmer)
                return(e.getSal()*0.1);
            else
                return 0;
    }
}
```

Output:

```
printing employee details

100  Joe  20000 tax paid :0.0

200  suhas  30000  IT  5 tax paid :6000.0
```

```
300  ajay  25000   3   java tax paid :2500.0

301  varun  25000   4   python tax paid :2500.0

=============================================================================================
```

Main:

```
package com.torryharris.mainpack;

import com.torryharris.emppack.employee;
import com.torryharris.emppack.manager;
import com.torryharris.emppack.programmer;

public class Main {

    public static void main(String[] args) {
   // write your code here
        manager m1 = new manager(200,"suhas",40000,"HR",5);
        System.out.println(m1+"  "+m1.calcNetsal());

        programmer p1 = new programmer(300,"ajay",30000,3,"java");
        System.out.println(p1+" "+p1.calcNetsal());

        employee[] emparr = new employee[4];
        emparr[0] = new manager(100,"joel",30000,"finance",6);
        emparr[1] = new manager(200,"suhas",30000,"IT",5);
        emparr[2] = new programmer(300,"ajay",25000,3,"java");
        emparr[3] = new programmer(301,"varun",25000,4,"python");

        System.out.println("printing employee details");
        for(employee e:emparr)
        {
            System.out.println(e+" --Net Salary: "+e.calcNetsal());
        }

    }
}
```

Employee class(abstract):

```
package com.torryharris.emppack;

public abstract class employee {
    protected int empid;
    protected String empname;
    protected int sal;


    public employee(int empid, String empname, int sal) {
        this.empid = empid;
        this.empname = empname;
```

```
        this.sal = sal;
    }
    public String getDetails()
    {
        return(empid+"  "+empname+"  "+sal);
    }
    public abstract double calcNetsal();
    public int getSal(){
        return sal;
    }


}
```

Manager class (sub):

```
package com.torryharris.emppack;

public class manager extends com.torryharris.emppack.employee {
    private String depname;
    private int empcount;

    public manager(int empid, String empname, int sal,String depname,int empcount) {
        super(empid, empname, sal);
        this.depname=depname;
        this.empcount=empcount;
    }

    @Override
    public String toString() {
        return "manager{" +
                "empid=" + empid +
                ", empname='" + empname + '\'' +
                ", sal=" + sal +
                ", depname='" + depname + '\'' +
                ", empcount=" + empcount +
                '}';
    }

    @Override
    public double calcNetsal() {
        return 2000;
    }

    public String getDepname(){
        return depname;
    }
    public int getEmpcount(){
        return empcount;
    }
}
```

Programmer class (sub):

```
package com.torryharris.emppack;

public class programmer extends com.torryharris.emppack.employee {
    private int noofproj;
    private String skillset;

    public programmer(int empid, String empname, int sal,int noofproj,String skillset) {
        super(empid, empname, sal);
        this.noofproj=noofproj;
        this.skillset=skillset;

    }

    @Override
    public String toString() {
        return "programmer{" +
                "empid=" + empid +
                ", empname='" + empname + '\'' +
                ", sal=" + sal +
                ", noofproj=" + noofproj +
                ", skillset='" + skillset + '\'' +
                '}';
    }

    @Override
    public double calcNetsal() {
        return 15000;
    }
}
```

Output:

```
manager{empid=200, empname='suhas', sal=40000, depname='HR', empcount=5}  2000.0

programmer{empid=300, empname='ajay', sal=30000, noofproj=3, skillset='java'} 15000.0

printing employee details

manager{empid=100, empname='joel', sal=30000, depname='finance', empcount=6} --Net
Salary: 2000.0

manager{empid=200, empname='suhas', sal=30000, depname='IT', empcount=5} --Net Salary:
2000.0

programmer{empid=300, empname='ajay', sal=25000, noofproj=3, skillset='java'} --Net
Salary: 15000.0

programmer{empid=301, empname='varun', sal=25000, noofproj=4, skillset='python'} --Net
Salary: 15000.0

================================================================================
```

Main:

```
package com.torryharris.mainpack;

import com.torryharris.pack1.class1;

public class Main {

    public static void main(String[] args) {
   // write your code here
        class1 ob = new class1();
        ob.m1();
        ob.m4();
    }
}
```

Interface1:

```
package com.torryharris.pack1;

public interface interface1 {
    void m1();
    void m2();
    void m3();
}
```

Interface2:

```
package com.torryharris.pack1;

public interface interface2 {
    int count=10;
    void m4();
}
```

Interface3:

```
package com.torryharris.pack1;

public interface interface3 extends interface1,interface2{

    @Override
    default void m1() {

    }

    @Override
    default void m2() {

    }
```

```
    @Override
    default void m3() {

    }

    @Override
    default void m4() {

    }
}
```

Class1:

```
package com.torryharris.pack1;

public class class1 implements interface1,interface2 {

    @Override
    public void m1() {
        System.out.println("in m1");

    }

    @Override
    public void m2() {
        System.out.println("in m2");


    }

    @Override
    public void m3() {
        System.out.println("in m3");


    }

    @Override
    public void m4() {
        System.out.println("in m4");
        System.out.println("count value "+interface2.count);


    }
}
```

output:

```
in m1
```

in m4

count value 10

==========

Main:

```
package com.torryharris.mainpack;

import com.torryharris.vpack.car;

public class Main {

    public static void main(String[] args) {
   // write your code here
        car car1 = new car(9938,"f ecosport",40,180,"sportsuv");
        System.out.println(car1.start());
        int increvalue = car1.increasespeed(140);
        if(increvalue==-1)
        {
            System.out.println("alert!!  stop the car");
        }
        else
            System.out.println("enjoy the drive");
        System.out.println(car1.stop());

    }
}
```

Automobile(interface):

```
package com.torryharris.vpack;

public interface Automobile {

    String start();
    int increasespeed(int n);
    String stop();

}
```

Vehicle(abstract class):

```
package com.torryharris.vpack;

public abstract class vehical implements Automobile {

    protected int regno;
    protected  String model;
    protected int currspeed;
```

```
    public vehical(int regno, String model, int currspeed) {
        this.regno = regno;
        this.model = model;
        this.currspeed = currspeed;
    }

    @Override
    public String start() {
        return "vehical started";
    }

    @Override
    public abstract int increasespeed(int n);


    @Override
    public String stop() {
        return "vehical stoped";
    }

    @Override
    public String toString() {
        return "vehical{" +
                "regno=" + regno +
                ", model='" + model + '\'' +
                ", currspeed=" + currspeed +
                '}';
    }
}
```

Car class:

```
package com.torryharris.vpack;

public class car extends vehical{
    private int maxspeed;
    private String type;


    public car(int regno,String model,int currspeed,int maxspeed,String type){
        super(regno,model,currspeed);
        this.maxspeed=maxspeed;
        this.type=type;
    }

    @Override
    public int increasespeed(int n) {
        if (currspeed+n < maxspeed)
        {
            currspeed += n;
        }
```

```
        else
        {
            currspeed=-1;
        }
        return (currspeed);
    }

    @Override
    public String toString() {
        return "car{" +
                "maxspeed=" + maxspeed +
                ", type='" + type + '\'' +
                ", regno=" + regno +
                ", model='" + model + '\'' +
                ", currspeed=" + currspeed +
                '}';
    }
}
```

Output:

```
vehical started

alert!!  stop the car

vehical stoped
```

==========