JAVA

**package** kaboos;

**public** **class** abdalla {

**public** **static** **void** main(String[] args) {

System.***out***.println("kaboos is coming");

System.***out***.println("\n AbdallaHijjawe");

}}

الشكل العام

**int** age=5;

**double** pi=2.5;

System.***out***.println("pi:"+pi+"age:"+age);

مثال ع تعريف المتغيرات

**import** java.util.Scanner;

**public** **class** abdalla {

**static** Scanner *reader* = **new** Scanner(System.***in***);

**public** **static** **void** main(String[] args) {

String name;

**int** age;

System.***out***.println("enter your name: ");

name=*reader*.nextLine();

System.***out***.println("enter your age: ");

age=*reader*.nextInt();

System.***out***.println("your age is:"+age + " your name is :"+name);

مثال على عمليه الادخال

**int** num;

String pho="553213";

num= Integer.*parseInt*(pho);

System.***out***.println(num);🡺553213

مثال ع طريقة التحويل من سترنغ الى انتجر

**int** num=20;

String pho="553213";

pho=String.*valueOf*(num);

System.***out***.println(pho);🡺20

تحويل من انتجر الى سترنغ

**int** age;

age = *reader*.nextInt();

**if**(age>18) {

System.***out***.println("your man");

مثال ع الجملة الشرطيه

**int** age;

age = *reader*.nextInt();

**if**(age>18) {

System.***out***.println("your man");}

**else** **if**(age<18&&age>10) {

System.***out***.println("your yong");}

**else** {

System.***out***.println("child");}

مثال اخر عن الجمل الشرطيه اكثر احترافيه

**for**(**int** i=0;i<5;i++)

{

System.***out***.println(i);🡺01234

}

شكل اللوب

**int** i=0;

**while**(i<10)

{System.***out***.println(i);

i++;}

**int** i=0;

**do**

{

System.***out***.println(i);

i++;

}**while**(i<10);

**int** i=0;

**int[]** array=**new** **int**[4];

array[0]=15;

array[1]=20;

الشكل العام للمصفوفه

**int** [] arr = {10,30,40,50};

**int** i=0;

**int** array[]=**new** **int**[4];

array[0]=15;

array[1]=20;

**for**(;i<5;i++) {

array[i]=*reader*.nextInt();

System.***out***.println(array[i]);}

شكل اخر للمصفوفه

**import** java.util.ArrayList;

ArrayList<String> name =**new** ArrayList<String>();

name.add("abdalla");

name.add("hijjawe");

name.add("kaboos");

System.***out***.println(name);🡺abdallahijjawekaboos

شكل العام لقائمة المصفوفه

ArrayList<String> name =**new** ArrayList<String>();

name.add("abdalla");

name.add("hijjawe");

name.add("kaboos");

System.***out***.println(name.get(1);🡺hijjawe

Name.contains(“ahmed”)🡺 هل الاسم موجود

ArrayList<String> abdalla =**new** ArrayList<String>();

abdalla.add("abdallaabdalla");

abdalla.add("hijjaweabdallla");

abdalla.add("kaboosabdlla");

ArrayList<String> hijjawe =**new** ArrayList<String>();

hijjawe.add("abdallahijjawe");

hijjawe.add("hijjawehijjawe");

hijjawe.add("kabooshijjawe");

ArrayList<String> kaboos =**new** ArrayList<String>();

kaboos.add("abdallakaboos");

kaboos.add("hijjawekaboos");

kaboos.add("kabooskaboos");

HashMap<String,ArrayList<String>> s=**new** HashMap<String,ArrayList<String>>();

s.put("abdallahijjawe", hijjawe);

System.***out***.println(s.get("abdallahijjawe")); 🡺رح يطبع لست حجاوي كامله

الشكل العام عن الهاش تيبل

**static** **void** fun2() {

**int** i=0;

**int** array[]=**new** **int**[4];

array[0]=15;

array[1]=20;

**for**(;i<5;i++) {

array[i]=*reader*.nextInt();

System.***out***.println(array[i]);}

}

**public** **static** **void** main(String[] args) {

*fun2*();

}

مثال عام عن الفنكشين

**static** **int** sum(**int** n1,**int** n2) {

**return**(n1+n2);

}

**static** **int** sum(**int** n1,**int** n2,**int** n3) {

**return**(n1+n2+n3);

}

**public** **static** **void** main(String[] args) {

**int** r2 = **new** abdalla().*sum*(5, 10);

**int** r3 = **new** abdalla().*sum*(10, 30, 20);

System.***out***.println(r2+ "\n"+ r3);

}

Function overloading

**void** display(**int** n1) {

System.***out***.println("integer"+n1);

}

**void** display(**double** n1) {

System.***out***.println("float"+n1);

}

**public** **static** **void** main(String[] args) {

**new** abdalla().display(1.5);}

# ex about Polymorphism

**final** **int** a=15;🡺 جعل للمتغير قيمه ثابته غير قابله للتعديل

**try** {

**int** n=*reader*.nextInt();

System.***out***.println(n);

}

**catch**(InputMismatchException ex) {

System.***out***.println("input should be int");

}

System.***out***.println("app is done");

}

يقوم الكود التالي بعمل تخطي اذا حدث خطأ فمثلا لو ادخل المستخدم قيمه غير انتجر يقوم بطباعه الجمله ويكمل الكود بشكل طبيعي

Catch(Exception ex)🡺 تقوم ع التقاط جميع الاخطاء

**try** {

FileWriter fout =**new** FileWriter("test.txt");

fout.write("abdallahijjawe is here");

fout.close();

}

**catch**(Exception ex){

System.***out***.println("erorr");}

انشاء ملف للطباعه

OOP

لاستدعاء الكلاس نقوم بالتالي

Name class anyname = new name class;

حتى اصل الى خصائص الكلاس بعد الاستدعاء

Nameclass.الخاصيه=any write

بعد الاستدعاء ممكن يكون في استدعاء سريع طريقه عمله نذهب الى الكلاس

Nameclass(){any code}

# وسيطبق تلقائيا ويسمى بالانجليزي constructor

# Ex:

**public** **static** **void** main(String[] args) {

hijjawe anyname=**new** hijjawe("kaboos","omar");

String Fullname= anyname.getname();

System.***out***.print(Fullname);

**public** **class** hijjawe {

String firstname;

String lastname;

hijjawe(String firstname,String lastname){

**this.**firstname= firstname;

**this.**lastname=lastname;

}

String getname() {

**return**(firstname+" "+lastname);

}

# Encapsulation🡺 متغيرات مخفيه لا يمكن الوصول اليها من خارج الكلاس

Ex:private String anyname;🡺لايمكن استدعاءها بكلاس اخر

# Inheritance🡺الوراثه

# Ex:

**public** **class** hijjawe **extends** university{

String firstname;

String lastname;

String getname() {

**return**(firstname+" "+lastname+" "+ "student in:"+uname );

**public** **class** university {

String uname;

**private** String uid;

**public** **class** abdalla {

**public** **static** **void** main(String[] args) {

hijjawe project =**new** hijjawe();

project.firstname="omar";

project.lastname="hijjawe";

project.uname="bau";

System.***out***.println(project.getname());

Ex **override:** **public** **class** abdalla {

**public** **static** **void** main(String[] args) {

hijjawe project =**new** hijjawe();

project.firstname="omar";

project.lastname="hijjawe";

project.uname="bau";

System.***out***.println(project.getname());🡺omar hijjawe

**public** **class** university {

String uname;

**private** String uid;

String getname() {

**return**(uname);

**public** **class** hijjawe **extends** university{

String firstname;

String lastname;

String getname() {

**return**(firstname+" "+lastname+" " );

Ex aggregation:

**public** **class** abdalla {

**public** **static** **void** main(String[] args) {

university mypro = **new** university();

mypro.uname="bau";

mypro.hij.firstname="omar";

mypro.hij.lastname="hijjawe";

System.***out***.println(mypro.getname());

**public** **class** university {

String uname;

hijjawe hij=**new** hijjawe();

String getname() {

**return**(uname+" "+hij.fullname());

**public** **class** hijjawe {

String firstname;

String lastname;

String fullname() {

**return**(firstname+" "+lastname);

**EX Enums:**

**public** **class** abdalla {

**public** **static** **void** main(String[] args) {

hijjawe pro =**new** hijjawe("abdalla","hijjawe",gender.***male***);

pro.display();

**public** **class** hijjawe {

String firstname;

String lastname;

gender type;

hijjawe(String firstname,String lastname,gender type){

**this**.firstname=firstname;

**this**.lastname=lastname;

**this**.type=type;

}

**void** display() {

System.***out***.println(firstname +" "+lastname+" "+type);

**public** **enum** gender {

***male***,

***female***

Ex interface:

**public** **class** hijjawe **implements** info {

String firstname;

String lastname;

@Override

**public** String getname() {

**return**(firstname+" "+lastname);

**public** **interface** info {

String getname();

**public** **class** university **implements** info {

String uname;

@Override

**public** String getname() {

**return**(uname);

}

**public** **class** abdalla {

**public** **static** **void** main(String[] args) {

hijjawe mys=**new** hijjawe();

mys.firstname="abdalla";

mys.lastname="hijjawe";

System.***out***.println(mys.getname());

}

# abstract 🡺 برنامج يجب ان يورث حتى يستخدم

# EX abstract:

**public** **class** abdalla {

**public** **static** **void** main(String[] args) {

hijjawe mys=**new** hijjawe();

mys.cardid="3004";

mys.code="1111";

mys.date="10/10";

System.***out***.println(mys.getname());

**public** **class** hijjawe **extends** cridetcard {

String firstname;

String lastname;

String getname() {

**return**(code+date+cardid);

**public** **abstract** **class** cridetcard {

String cardid;

String date;

String code;}

Ex this:

**public** **class** abdalla {

**public** **static** **void** main(String[] args) {

bike vb= **new** bike();

vb.getspeed();🡺40

**public** **class** veicle {

**int** speed=60;

String modle;

**public** **class** bike **extends** veicle {

**int** speed=40;

**void** getspeed() {

System.***out***.println(**this**.speed);

change between (this) and super🡺 للحصول ع السرعه العامه

inner class🡺nested class

EX inner class:

**public** **class** abdalla {

**public** **static** **void** main(String[] args) {

car mycar=**new** car();

mycar.speed=20;

mycar.model="bmw";

System.***out***.println(mycar.model+mycar.speed);}

**static** **class** car{

**int** speed;

String model;}}

Import java.io.\*;🡺 هيك رح يختار كل الكلاسات الموجوده بالبكج

End oop

How work casting:

Double n1=5.5;

Int nicast=(int)n1;

Ex Casting in more class:

**public** **class** abdalla {

**public** **static** **void** main(String[] args) {

sebooks mybook=**new** sebooks();

mybook.section="14b";

mybook.serialnum=56465;

library mylab = (library)mybook;🡺هيك لاب رح يوخذ بس سكشين لانها الوحيده الموجوده بالمكتبه

System.***out***.prinln(mylab.section);

**public** **class** library {

String section;

**public** **class** sebooks **extends** library {

**int** serialnum;

thread🡺 نظريه بتحكي ان ثلاث عمليات بتشتغل مع بعض طبيعي فيديو 39من خلال مكتبه ثريد

# synchronized 🡺 بتخلي الثريد تطبع وحده وحده توضع قبل الثيورم

# deadlock🡺 ثريد1 وثريد2 لو كل وحده بدها توصل لاوبجيكت الثانيهرح يضلهن يستنتن بعض ومش رح يوصلن

# Join🡺 بتعمل توقف للبرنامج حتى يكتمل الثريد

# Lambda

# Name interface instance new name =()->{any code};

**public** **class** abdalla {

**public** **static** **void** main(String[]args) {

ArrayList<login> myuser = **new** ArrayList<login>();

myuser.add(**new** login("abdalla","123456"));

myuser.forEach(user->{

System.***out***.println("user"+user.user);

System.***out***.println("password"+user.password);

});

# Predicate<login> prerules=(p)->p.weakpassword()==true;

**If(prerules.test(user)){** System.***out***.println("user"+user.user);

System.***out***.println("password"+user.password);

# }

# Public boolean weakpassword(){

# If(password.length()<4)

# Return true

# Else

# Return false}

# رح يطبع المستخدمين يلي حققو الشرط