**JAVA MISSED QUIZZES**

**Java control statements:**

1. public class IfElse {

public static void main(String[] args) {

int number=13;

if(number%2==0){

System.out.println("even number");

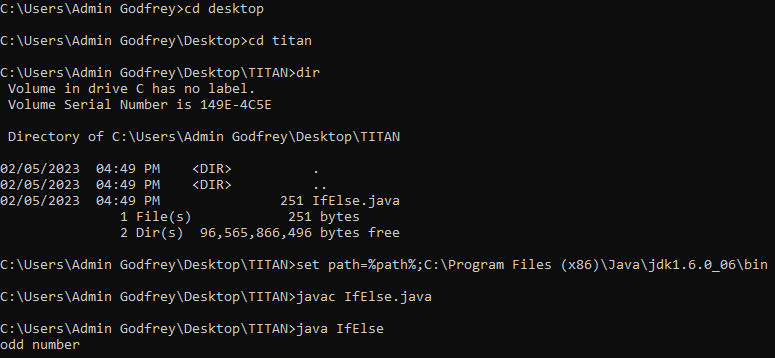
}else{

System.out.println("odd number");

}

}

}



2. public class SwitchExample {

public static void main(String[] args) {

int number=20;

switch(number){

case 10: System.out.println("10");

break;

case 20: System.out.println("20");

break;

case 30: System.out.println("30");

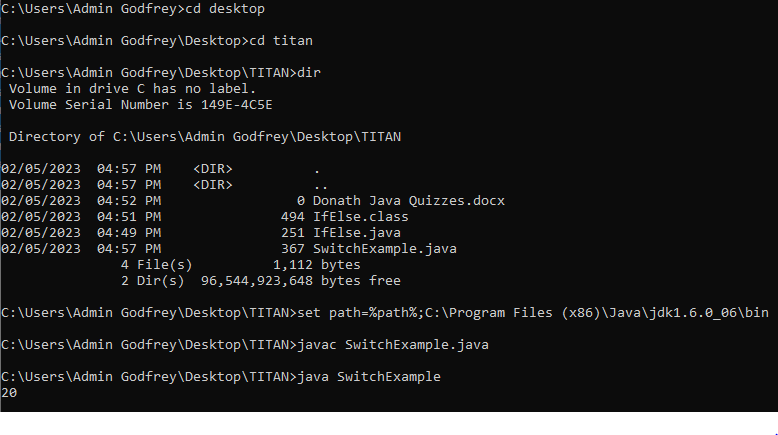
break;

default:System.out.println("Not in 10, 20 or 30");

}

}

}



3. public class PyramidExample {

public static void main(String[] args) {

for(int i=1;i<=5;i++){

for(int j=1;j<=i;j++){

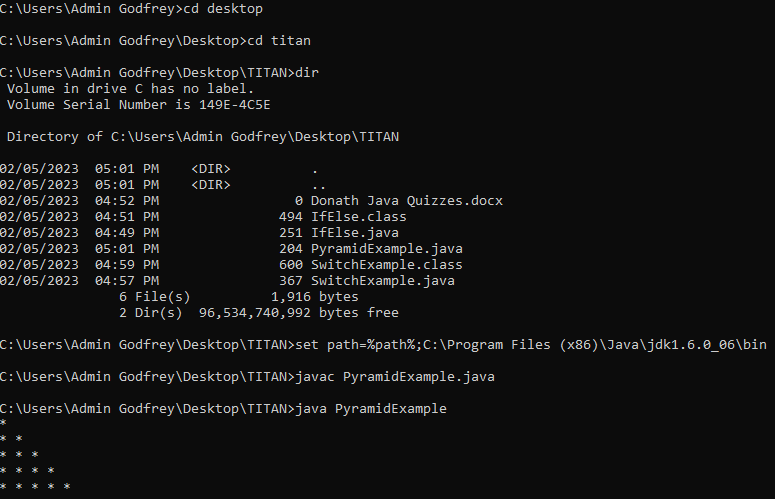
System.out.print("\* ");

}

System.out.println();

}

}

} 

**JAVA OBJECT CLASS:**

1. class Student{

int id;

String name;

}

class TestStudent2{

public static void main(String args[]){

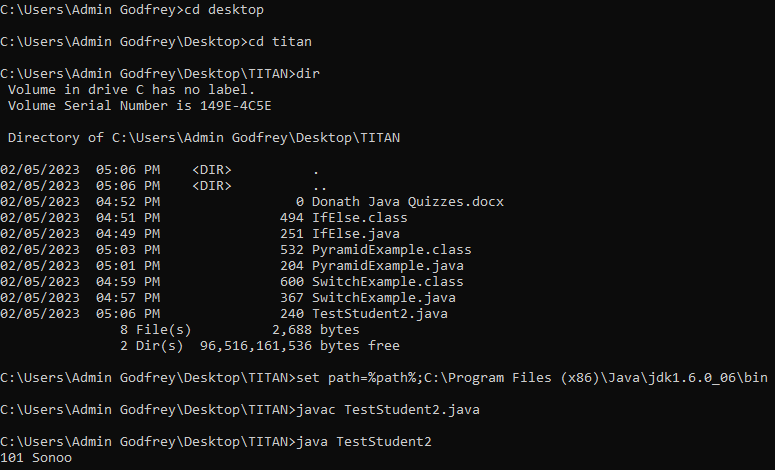
Student s1=new Student();

s1.id=101;

s1.name="Sonoo";

System.out.println(s1.id+" "+s1.name);

}

} 

2. public class Demo

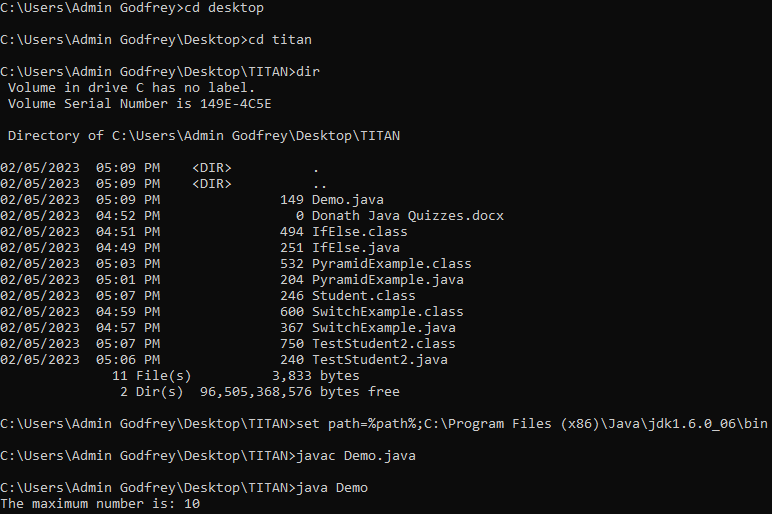
{

public static void main(String[] args)

{

System.out.print("The maximum number is: " + Math.max(10,7));

}

} 

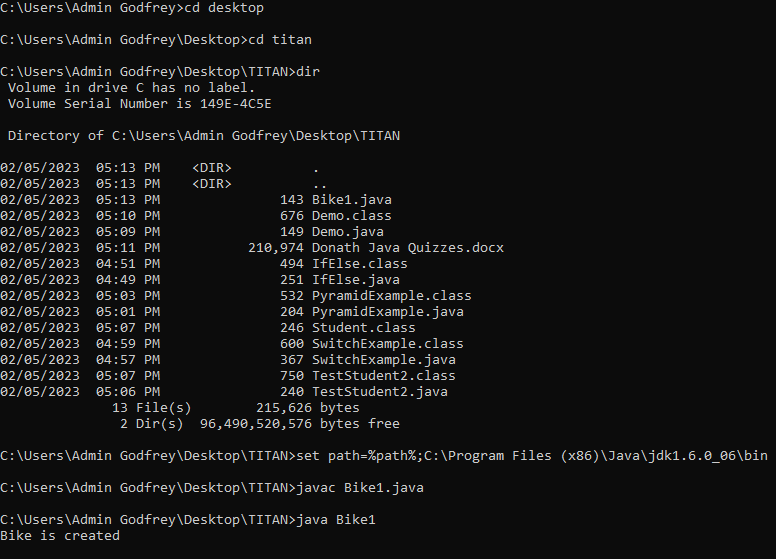
3. class Bike1{

Bike1(){System.out.println("Bike is created");}

public static void main(String args[]){

Bike1 b=new Bike1();

}

} 

**JAVA INHERITANCE:**

1. class Employee{

float salary=40000;

}

class Programmer extends Employee{

int bonus=10000;

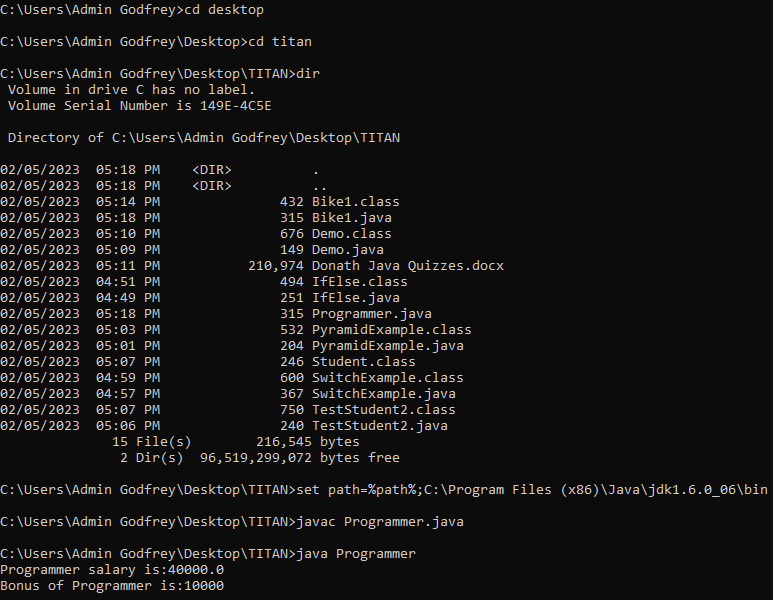
public static void main(String args[]){

Programmer p=new Programmer();

System.out.println("Programmer salary is:"+p.salary);

System.out.println("Bonus of Programmer is:"+p.bonus);

}

} 

2. class Animal{

void eat(){System.out.println("eating...");}

}

class Dog extends Animal{

void bark(){System.out.println("barking...");}

}

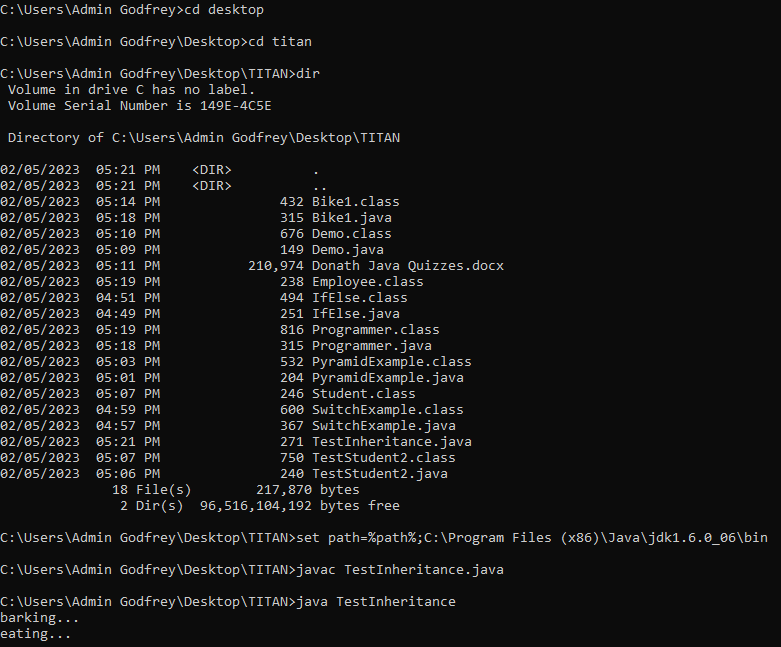
class TestInheritance{

public static void main(String args[]){

Dog d=new Dog();

d.bark();

d.eat();

}} 

3. class Operation{

int square(int n){

return n\*n;

}

}

class Circle{

Operation op;

double pi=3.14;

double area(int radius){

op=new Operation();

int rsquare=op.square(radius);

return pi\*rsquare;

}

public static void main(String args[]){

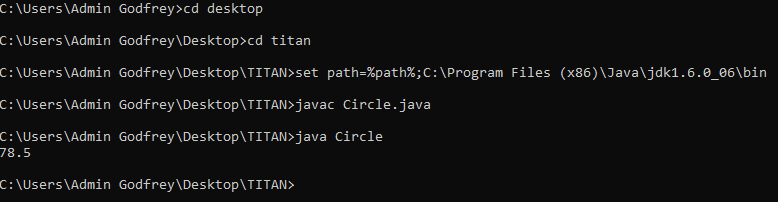
Circle c=new Circle();

double result=c.area(5);

System.out.println(result);

}

}



**JAVA POLYMORPHISM:**

1. class Adder{

static int add(int a,int b){return a+b;}

static int adder(int a,int b,int c){return a+b+c;}

}

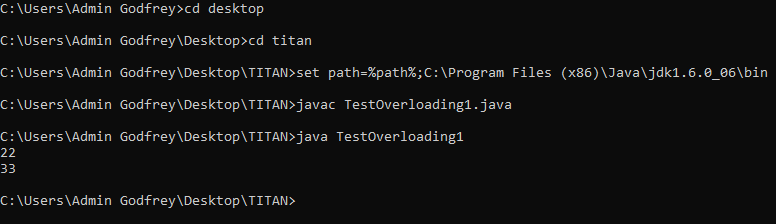
class TestOverloading1{

public static void main(String[] args){

System.out.println(Adder.add(11,11));

System.out.println(Adder.adder(11,11,11));

}}



2. class Vehicle{

void run(){System.out.println("Vehicle is running");}

}

class Bike extends Vehicle{

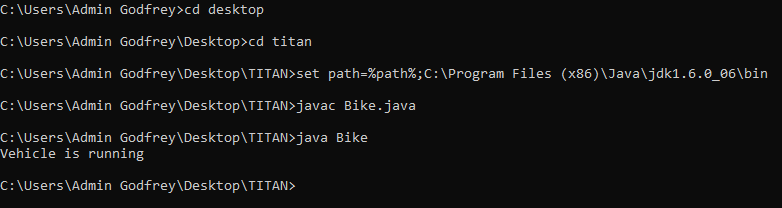
public static void main(String args[]){

Bike obj = new Bike();

obj.run();

}

}



3. class A{

A get(){return this;}

}

class B1 extends A{

@Override

B1 get(){return this;}

void message(){System.out.println("welcome to covariant return type");}

public static void main(String args[]){

new B1().get().message();

}

}

