Set 3

1)

public class ConcatTest{

static String concatWithString() {

String t = "Java";

for (int i=0; i<10000; i++){

t = t + "set";

}

return t;

}

static String concatWithStringBuffer(){

Stringbuffer sb = new Stringbuffer("Java");

for (int i=0; i<10000; i++){

sb.append("set");

}

return sb.toString();

}

public static void main(String args){

long startTime = System.currentTimeMillis()

concatWithString();

System.out.println("Time taken by Concating with String: "+(System.currentTimeMillis()-starttime)+"ms");

startTime = System.currentTimeMillis();

concatWithStringBuffer();

System.out.println("Time taken by Concating with StringBuffer: "+(System.currentTimeMillis()-starttime)+"ms");

}

}

Errors: pulic methods,starttime,semicolon , main identifier

2)

class Student{

void insertRecord(int r, String n){

rollno=r;

name=n;

}

void displayInformation()

{System.out.println(rollno+" "+name);}

}

public static void main(String args[]){

Student s1= Student();

Student s2= Student();

s1.insertRecord(111,"Karan");

s2.insertRecord(222,"Aryan");

s1.displayInformation();

s2.displayInformation();

}

3)

class vehicle

{

int s;d;t;

vehicle()

{

System.out.println("vehicle class");

}

void speed(intd,int t)

{

s=d/t;

System.out.println("speed:"+s);

}

}

class veh1 extend vehicle

{

s1,d1,t1;

veh1()

{

System.out.println("inheritance veh1 class");

}

void speed(double d1,double t1)

{

s1=d1/t1;

System.out.println("speed:"+s1);

}

}

public class heirarchypolymorphism

{

public static void main(String args[])

{

veh1 v=veh1();

v.speed(10,5);

v.speed(10.5,5.2);

}

}

Errors:

Semicolon In int ,extend,double declaration ,main,new

4)

class A

{

int x=10;

void display()

{

System.out.println("x="+x);

}

}

class B

{

int y=20;

void display1();

}

class C extends A implements B

{

public void display1()

{

System.out.println("y="+y);

}

}

class Multi

{

main(String args[])

{

c obj=new C()

obj.display()

obj.display1()

}

Errors:

Main,obj declaration,interface,parse,semicolons

5)

package jdbcdemo;

public class jdbcdemo1 {

public static void main(String args[]) {

try {

Class.forName("oracle.driver.OracleDriver");

// System.out.println("driver i registered with driver manager");

Connection cn=driverManager.getconnection("jdbc:oracle:thin:@172.20.0.14:1521:CSE201718IISEM", "cse\_022", "cse\_022");

System.out.println("connection was established");

Statement ps=cn.createStatement();

// ps.execute("drop table student");

ps.executeUpdate("create table student(rollno NUMBER(), name VARCHAR2(), marks NUMBER())");

// int result=ps.executeUpdate("create table student(rollno NUMBER(3), name VARCHAR2(10), marks NUMBER(10,3),CONSTRAINT indent PRIMARY KEY(rollno))");

ps.executeUpdate("insert into student (82,'vanitha',9.16)");

ps.executeUpdate("insert into student (64,'charitha',9.16)");

ps.executeUpdate("insert into student (66,'hyndavi',9.16)");

ps.executeUpdate("insert into student (76,'nikhita',9.16)");

int result=ps.executeUpdate(null);

System.out.println(result);

}

catch e)

{

System.out.println("CLASS NOT FOUND");

}

catch( e)

{

System.out.println("error in driver loading");

}

}

}

Errors:import stmts,jdbc (connection),driver ,connection,datatype values,insert stmt,exceptions

Set4

1) import java.util.Scanner;

public class lab10

{

main(String args[])

{

int age;

String s;

scanner o=new scanner(system.in);

try

{

try

{

System.out.println("enter the age");

age=o.nextInt();

if(age>50)

throw ArithmeticException("age should not be greater than 50");

}

catch(ArithmeticException )

{

System.out.println(e.getMessage());

}

System.out.println("enter the String");

s=o.next();

if(!(s.matches("[a-zA-Z]+")))

{

throw Exception();

}

else

println("string is:"+s);

}

catch(Exception )

{

System.out.println("caught the name exception");

}}

Errors:

Exceptions,throw stmnts,scanner,system,main,parse errors