1. FIND THE OUTPUT:

class Employee{

public void fun(){

System.out.println("Employee");

}

}

class Manager extends Employee{

public void fun(){

System.out.println("Manager");

}

}

class Method{

public static void main(String args[]){

Employee m=new Manager();

Manager e=new Manager();

m=e;

e.fun();

m.fun();

}

}

a) Manager

Employee

b) Employee

Manager

c) Error

d) Manager

Manager

2. FIND THE OUTPUT:

class access

{

static int x;

void increment()

{

x++;

}

}

class static\_use

{

public static void main(String args[])

{

access obj1 = new access();

access obj2 = new access();

obj1.x = 0;

obj1.increment();

obj2.increment();

System.out.println(obj1.x + " " + obj2.x);

}

}

a) 1 2

b) 1 1

c) 2 2

d) Compilation Error

3. FIND THE OUTPUT:

class Weather

{

public static void main(String[] args)

{

throw new NullPointerException("Hello");

}

catch(ArithmeticException e)

{

System.out.println("B");

}

}

a)Compile time error

b)0

c)B

d)Run time error

4. FIND THE OUTPUT:

class control\_statments

{

public static void main(String args[])

{

int x = 2;

int y = 0;

for ( ; y < 10; ++y)

{

if (y % x == 0)

continue;

else if (y == 8)

break;

else

System.out.print(y + " ");

}

}

}

a)2

b)3

c)5

d)error

5. FIND THE OUTPUT:

public class MapTest

{

public static void main(String args[])

{

Map m = new HashMap();

m.put(null,"Test");

m.put(null,"Fest");

System.out.println(m);

}

}

a) {null=Fest}

b) null

c) [null=Fest]

d) {null=Test}

6.

class A{

String args[] = { "1" , "2" };

public static void main(String args[]) {

if (args.length >0)

System.out.println(args.length);

}

}

options

a. the program compiles but print nothing.

b. compile error.

c. 2.

d. 0.

7.

class E{

int empid=101;

public void opt1(){

System.out.println("hello");

}

}

class V extends E{

String empname="xyz";

public void opt1(){

System.out.println("hi");

}

public static void main(String[] args){

E e = new V();

V v = (V)e;

v.opt1();

System.out.println("Id is:"+v.empid);

System.out.println("Name is:"+v.empname);

}

}

options

a. compilation error.

b. hello

Id is:101

Name is:xyz.

c. hi

Id is:101

Name is:xyz.

d. hello

hi

Id is:101

Name is:xyz.

8.

class Staticsamp{

static int empid = 10;

static String empname = "hai";

public static void sample(){

System.out.println("empid:" + empid + "\n" + "empname:" + empname);

}

static{

System.out.print("hello");

}

Staticsamp(){

System.out.println("the output is");

}

public static void main(String[] args){

Staticsamp s = new Staticsamp();

sample();

}

}

options

a.compile error.

b.hellothe output is

empid: 10

empname:hai

c.Runtime error.

d.hello

the output is

empid: 10

empname:hai

9.

import java.util.\*;

class JavaCollectionExample

{

public static void main(String args[])

{

Hashtable<String, Double> balance= new Hashtable<String, Double>();

Enumeration<String> names;

String str;

double bal;

balance.put("Deepak Patel", 3243.53);

names = balance.keys();

while(names.hasMoreElements())

{

str = names.nextElement();

System.out.println(str + ": " + balance.get(str));

}

System.out.println();

bal = balance.get("Deepak Patel");

balance.put("Deepak Patel", bal+1000);

System.out.println("Deepak Patel's new balance: " + balance.get("Deepak Patel"));

}

}

options

(a)Deepak Patel: 3243.53

Deepak Patel's new balance: 4243.530000000001

(b)Deepak Patel: 3243.53

Deepak Patel's new balance: 4243.53

(c)Deepak Patel: 3243.53000000000001

Deepak Patel's new balance: 4243.530000000001

(d)Deepak Patel: 3243.53

Deepak Patel's new balance: 4243.5300001

10.

import java.io.IOException;

class Testthrows1{

void m()throws IOException{

throw new IOException("device error");//checked exception

}

void n()throws IOException{

m();

}

void p(){

try{

n();

}catch(Exception e){System.out.println("exception handled");}

}

public static void main(String args[]){

Testthrows1 obj=new Testthrows1();

obj.p();

System.out.println("normal flow");

}

}

options

a.error

b.exception handled

normal flow

c.exception handled

d.input mismatch token.

11.class switchcase {

public static void main(String[] args) {

int x = 100, y = 101;

switch (x & y){

case 0:

System.out.println("India");

case 1:

System.out.println("USA");

case 100:

System.out.println("China");

case 101:

System.out.println("Russia");

default:

System.out.println("Default "+x+" "+y);

}

}

}

A. Error

B. China

Russia

Default 100 101

C. India

D. USA

12.

class Str {

public static void main(String[] args) {

String a = "abc";

String ab = a + "def";

String abc = "abcdef";

if (ab == abc) {

System.out.println("True "+abc);

}

else{

System.out.println("false "+ab);

}

if (ab.equals(abc)){

System.out.println("True "+abc);

}

else{

System.out.println("false "+ab);

}

}

}

A.

false abcdef

True abcdef

B.

false abcdef

false abcdef

C.

True abcdef

True abcdef

D.

True abcdef

false abcdef

13.

import java.util.\*;

class a{

private int no;

private String str;

a(int x,String y)

{

no=x;

str=y;

}

private int geta()

{

return no;

}

private String gets()

{

return str;

}

}

class collections{

public static void main(String args[])

{

a a1 = new a(10,"A");

a a2 = new a(20,"B");

a a3 = new a(30,"C");

a a4 = new a(40,"D");

a a5 = new a(50,"E");

ArrayList<a> l =new ArrayList<a>();

l.add(a1);

l.add(a2);

l.add(a3);

l.add(a4);;;

l.add(a5);

Iterator<a> i = l.iterator();

while(i.hasNext()){

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

}

}

}

A [10=A, 20=B, 30=C, 40=D, 50=E]

B Garbage value

C Compilation Error

D [10,A 20,B 30,C 40,D 50,E]

14.

class Demo {

static int i =4 ;

public static void main(String[] args)

{

int i = 1;

for (Demo.i = 2; Demo.i < 10; Demo.i++) {

i = i + 2;

System.out.print(i + " ");

}

}

}

a)3 5 7 9 11 13 15 17

b)4 6 8 10

c)compiletime error

d)1 2 3 … 9

15.

class try\_catch{

public static void main(String args[])

{

int a=10,b=0,c;

try {

c=a/b;

}

catch ()

{

System.out.println("Exception Occured ");

}

finally

{

System.out.println("This is finally" );

}

}

}

A Arithmatic Exception in runtime

B Exception Occured

This is finally

C Compilation Error

D This is finally

1. What will be the output?

import java.util.\*;

class Customer

{

int cid;

String cname;

public Customer(int cid,String name)

{

this.cid=cid;

this.cname=name;

}

public void setCid(int cid)

{

this.cid=cid;

}

public int getCid()

{

return cid;

}

public void setName(String cname)

{

this.cname=cname;

}

public String getName()

{

return cname;

}

public static void main(String args[])

{

Customer cust=new Customer(11,"Ritika");

Customer cust1=new Customer(12,"Jyoti");

Customer cust2=new Customer(13,"divya");

ArrayList<Customer> list=new ArrayList<Customer>();

list.add(cust);

list.add(cust1);

list.add(cust2);

/\*ArrayList<Integer> l=new ArrayList<Integer>();

l.add(22);

l.add(33);\*/

Iterator<Customer> it1=list.iterator();

while(it1.hasNext())

{

System.out.println(it1.next());

}

}

}

1. Compilation error
2. 11 Ritika

12 Jyoti

13 divya

1. Garbage value
2. What will be the output?

class employee

{

public void fun()

{

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

}

}

class Manager extends employee

{

public void fun1()

{

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

}

}

class Organization5

{

public static void main(String args[])

{

employee e=new Manager();

e.fun1();

}

}

a)compilation error

b)child class

c)parent class

d)runtime error

18) Find the output.

class Club

{

    static String fn,ln;

    static int mem=0;

    public Club(String fr, String la)

    {

        fn=fr;

        ln=la;

        mem++;

    }

    public static void main(String[] args)

    {

        Club c=new Club("ms","d");

        Club c1=new Club("vir","seh");

        System.out.println(fn+ln+mem);

    }

}

a) msd1

   virseh2

b) msd1

c)virseh2

1. Error

19) Find the output.

class Base extends Exception {}

class Derived extends Base {}

public class Main {

public static void main(String args[]) {

try {

throw new Derived();

}

catch(Base b) {

System.out.println("Caught base class exception");

}

}

}

1. Caught base class exception
2. Caught derived class exception
3. Compiler Error because derived is not throwable
4. Compiler Error because base class exception is caught before derived class

20)Predict the output.

class Exam

{

    public static void main(String args[])

    {

        int i;

        for(i=0;i<10;i++);

        {

        if(i<5)

            {

                System.out.println("Test Passed");

            }

            else if(i<10)

            {

                System.out.println("Test Failed");

            }

            else

            {

                System.out.println("Withheld");

            }

        }

        switch(i)

        {

            case 0:System.out.println(i);break;

            case 1:System.out.println(i);break;

            case 2:System.out.println(i);break;

            case 3:System.out.println(i);break;

            case 4:System.out.println(i);break;

            case 5:System.out.println(i);break;

            case 6:System.out.println(i);break;

            case 7:System.out.println(i);break;

            case 8:System.out.println(i);break;

            case 9:System.out.println(i);break;

            default:System.out.println("Completed");break;

        }

    }

}

A.      Withheld

Completed

B.      Test Failed  
  
9

C.      Test Failed  
  
Completed

D.      Syntax Error

21. public class CppBuzz{

public static void main(String[] args){

int a = 5;

a +=5;

switch(a){

case 5: System.out.print("5");break;

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

System.out.println(((a%2 ==0) ? "-even-" : "-odd-"));

break;

default: System.out.print("0");

}

}

}

(A) Compilation Error

(B) 10-even-

(C) 10-even-0

(D) 10-odd

22. class Test1 {

static int x = 10;

public

static void main(String[] args)

{

Test1 t1 = new Test1();

Test1 t2 = new Test1();

t1.x = 20;

System.out.print(t1.x + " ");

System.out.println(t2.x);

}

}

A) 10 10

B) 20 20

C) 10 20

D) 20 10

23.public class Testing {

public int number = 1;

public String getColor(){

return "red";

}

}

public class Test extends Testing{

public int number = 2;

public String getColor(){

return "blue";

}

public static void main(String[] args){

Testing supersub = new Test();

Testing supersub1 = new Test();

System.out.println( supersub.getColor() + supersub1.number);

}

}

a. red1

b. red2

c.blue1

d.blue2

24.import java.io.\*;

import java.util.\*;

class Test

{

public static void main (String[] args)

{

int arr[] = new int[] {1, 2, 3, 4};

Vector<Integer> v = new Vector();

Hashtable<Integer, String> h = new Hashtable();

v.addElement(1);

v.add(2);

h.put(1,"FTP");

h.put(2,"4FTP");

System.out.println(arr[0]);

System.out.println(v.elementAt(1));

System.out.println(h.get(1));

}

}

a) 1

2

1,FTP

b) 1

2

4FTP

c)1

2

FTP

d) compilation error

25.class newclo

{ int count = 0;

void A() throws Exception

{

try

{

count++;

try

{

count++;

try

{

count++;

throw new Exception();

}

catch(Exception ex)

{

count++;

}

}

catch(Exception ex)

{

count++;

throw new Exception();

}

}

catch(Exception ex)

{

count++;

}

}

void display()

{

System.out.println(count);

}

public static void main(String[] args) throws Exception

{

newclo obj = new newclo();

obj.A();

obj.display();

}

}

a. 4

b. 5

c. 6

d. error

26) Find the output:

public class Demo

{

static

{

System.out.println("First static block");

}

public Demo()

{

System.out.println("Constructor");

}

public static String staticString = "Static Variable";

static

{

System.out.println("Second static block and "+ staticString);

}

static

{

staticMethod();

System.out.println("Third static block");

}

public static void staticMethod()

{

System.out.println("Static method");

}

public static void staticMethod2()

{

System.out.println("Static method2");

}

public static void main(String[] args)

{

Demo obj = new Demo();

obj.staticMethod2();

}

}

a)First static block

Second static block and Static Variable

Static method

Third static block

Constructor

Static method2

b)Static method

First static block

Second static block and Static Variable

Third static block

Constructor

Static method2

c)First static block

Constructor

Second static block and Static Variable

Static method

Third static block

Static method2

d)First static block

Second static block and Static Variable

Constructor

Static method

Third static block

Static method2

27) import java.io.\*;

class Parents{

public void fun(){

System.out.println("parent");

}

class TestExceptionChild extends Parents{

public void fun(){

System.out.println("TestExceptionChild");

}

public static void main(String args[]){

Parents o =new TestExceptionChild();

TestExceptionChild m =(TestExceptionChild) o;

m.fun();

}

}

A) TestExceptionChild

b) parent

c) Compile time Error

d) none

28. What is the output.

class CollectionIterator {

public static void main(String args[]) {

ListIterator a = list.ListIterator();

if(a.previousIndex()!= -1)

while(a.hasNext())

System.out.print(a.next() + " ");

else

System.out.print("Empty");

}

}

a) 0

b) error

c) -1

d) Empty

29. Output??

public class ThrowExample {

static void checkEligibilty(int stuage, int stuweight){

if(stuage<12 && stuweight<40) {

throw new ArithmeticException("Student is not eligible for registration.");

}

else {

System.out.println("Student Entry is Valid!!");

}

}

public static void main(String args[]){

System.out.println("Welcome to the Registration process!!");

checkEligibilty(10, 39);

System.out.println("Have a nice day..");

}

}

A)Welcome to the Registration process!!Exception in thread "main"

java.lang.ArithmeticException: Student is not eligible for registration

B)Student is not eligible for registration.Welcome to the Registration process!

C)Compile time error

D)Welcome to the Registration process!!Exception in thread "main"

java.lang.ArithmeticException: Student is not eligible for registration

Have a nice day..

30. What is the output

class ControlStatements {

public static void main(String args[]) {

int i =1;

do{

if(i==5){

i++;

break;

}

System.out.println(i);

i++;

}

while(i<=10);

}

}

a)1

b) 1

2

3

4

c)infinte loop

d)error