**class** StaticDemo {

**static** **int** *a*=0;

**int** b=++*a*;

**static** **int** *c*=++*a*;

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

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

}

}

What will be output of above program?

(a)0

(b)1

(c)2

(d)Compiler error

Answer: (b)

(2)

**class** StaticDemo {

**int** a=0;

**int** b=++a;

**static** **int** c=++a;

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

System.out.print(c^b);

}

}

What will be output of above program?

(a)0

(b)1

(c)2

(d)Compiler error

Answer: (d)

(3)

**package** manish;

**class** StaticDemo {

**static** {

try {

System.out.println("raja");

}

**catch**(Exception e){

}

}

}

What will be output of above program?

(a)raja

(b)No main method in **class** exception.

(c)raja

No main method in **class** exception.

(d)Compiler error

Answer: (c: for java6 and below else b for java7 onwards)

(4)

**class** StaticDemo {

**static** {

System.out.println("raja");

}

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

System.out.println("rani");

}

}

What will be output of above program?

(a)rani

raja

(b)raja

rani

(c)rani

(d)Compiler error

Answer: (b)

(5)

**class** StaticDemo {

**static** **int** a=0;

**static** {

System.out.println(a++);

}

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

++a;

System.out.println(++a);

}

**static** {++a;}

}

What will be output of above program?

(a)1

3

(b)0

4

(c)2

2

(d)Compiler error

Answer: (b)

class Square{

static double *pi*=Math.*PI*;

static void area(double r){

System.*out*.print((int)*pi*\*r\*r);

}

}

class Cube extends Square{

static void area(double r){

System.*out*.print(6\*(int)*pi*\*r\*r);

}

}

class SuperDemo extends Cube{

public static void main(String[] args) {

SuperDemo t=new SuperDemo();

t.area(2);

}

void area(int r){

super.*area*(r);

}

}

What will output when you compile and run the above code?

(a)12.0

(b)68.45

(c)72.0

(d)Compiler error

Answer: (c)

(2)

class World{

World where(){

System.*out*.println("I think about our world");

return this;

}

}

class CounSuperDemo extends World{

CounSuperDemo where(){

System.*out*.println("I think about our country");

return this;

}

}

class State extends CounSuperDemo{

State where(){

System.*out*.println("I think about our State");

return this;

}

}

class SuperDemo extends State{

public static void main(String[] args) {

SuperDemo t=new SuperDemo();

t.call();

}

void call(){

World w=super.where();

w.where();

}

}

What will output when you compile and run the above code?

(a) I think about our State

I think about our State

(b) I think about our country

I think about our State

(c) I think about our State

I think about our country

(d)Compiler error

Answer: (a)

(3)

class World{

World where(){

System.out.println("I think about our world");

return this;

}

}

class CounSuperDemo extends World{

World where(){

System.out.println("I think about our counSuperDemo");

return this;

}

}

class State extends CounSuperDemo{

CounSuperDemo where(){

System.out.println("I think about our State");

return this;

}

}

class SuperDemo extends State{

static SuperDemo t;

public static void main(String[] args) {

t=new SuperDemo();

t.call();

}

void call(){

((t.where()).where()).where();

}

}

What will output when you compile and run the above code?

(a) I think about our State

I think about our counSuperDemo

I think about our counSuperDemo

(b) I think about our State

I think about our State

I think about our State

(c) I think about our counSuperDemo

I think about our State

I think about our State

(d)Compiler error

Answer: (b)

(4)

class World{

World where(){

System.out.println("I think about our world");

return this;

}

}

class CounSuperDemo extends World{

World where(){

System.out.println("I think about our country");

Return new World();

}

}

class State extends CounSuperDemo{

World where(){

System.out.println("I think about our State");

return new CounSuperDemo();

}

}

class SuperDemo extends State{

public static void main(String[] args) {

CounSuperDemo c=new State();

c.where().where().where();

}

}

What will output when you compile and run the above code?

(a) I think about our World

I think about our country

I think about our state

(b) I think about our Country

I think about our state

I think about our world

(c) I think about our State

I think about our country

I think about our world

(d)Compiler error

Answer: (c)

(5)

class Sea{

int a,b;

Sea(int a,int b){

this.a=a;

this.b=b;

}

}

class SuperDemo extends Sea{

int a,b;

static SuperDemo t;

public static void main(String[] args) {

t=new SuperDemo(1,2,3,4);

t.find();

}

SuperDemo(int a,int b,int c,int d){

super(c,d);

this.a=a;

this.b=b;

}

void find(){

System.out.print(a+t.b+this.a+super.b);

}

}

What will output when you compile and run the above code?

(a)7

(b)8

(c)9

(d)Compiler error

Answer: (b)

**class** Square{ **static** **double** *pi*=Math.*PI*; **static** **void** area(**double** r){ System.*out*.print((**int**)*pi*\*r\*r); }}

**class** Cube **extends** Square{ **static** **void** area(**double** r){ System.*out*.print(6\*(**int**)*pi*\*r\*r); }}

**class** Test2 **extends** Cube{ **public** **static** **void** main(String[] args) {

Test2 t=**new** Test2(); t.area(2); }

@Override

**void** area(**double** r){ **super**.*area*(r); }}

Answer: Compile time error: cannot override a static method with non-static method.

**class** World{ World where(){ System.*out*.println("I think about our world"); **return** **this**; }}

**class** CounSuperDemo **extends** World{ CounSuperDemo where(){ System.*out*.println("I think about our country"); **return** **this**; }}

**class** State **extends** CounSuperDemo{ World where(){ System.*out*.println("I think about our State"); **return** **this**; }}

**class** Test2 **extends** State{ **public** **static** **void** main(String[] args) {

Test2 t=**new** Test2(); t.call(); }

**void** call(){ World w=**super**.where(); w.where(); }}

Answer: Compile time error: cannot override method with return type as superclass, must be a subclass or same class.

HYPERLINK "//www.blogger.com/rearrange?blogID=959939550345671470&widgetType=HTML&widgetId=HTML1&action=editWidget&sectionId=main"HYPERLINK "//www.blogger.com/rearrange?blogID=959939550345671470&widgetType=HTML&widgetId=HTML1&action=editWidget&sectionId=main"

**variables questions on java with explanation**

**Questions on variables in java with answers**

(1)

public class Test {

public static void main(String[] args) {

int a=5;

{

Integer b=10;

}

int c=a+b;

System.out.println(c);

}

}

What will output when you compile and run the above code?

(a)15

(b)5

(c)Compiler error

(d) Run time error

Answer: (c)

Explanation:

Scope of the any variable in a function is the braces in which it has been declared. Variable a has been declared within main function braces. So its scope is whole the main function while variable b has been declared in another braces so its scope and visibility is only that block or braces. After that block variable a is visible while variable b has dead.

So, It is error to write int c=a+b;

(2)

public class Loop {

public static void main(String[] args) {

for(int i=-0;i<3;i++){

System.out.print(i);

}

System.out.print(i);

}

}

What will output when you compile and run the above code?

(a) 0 1 2 3

(b) 0 1 2 2

(c) 0 0 1 2

(d) Compiler error

Answer: (d)

Explanation:

Scope of the any variable is in a function or loop or the only braces in which it has been declared. Variable i has been declared within for loop. So it is not visible to outside the for loop

Identifier

(3)

public class Identifier {

public static void main(String[] args) {

int a$=5;

Integer b$=10;

int \_c=a$+b$;

System.out.print(\_c);

}

}

What will output when you compile and run the above code?

(a) 15

(b) 50

(c) Compile time error

(d) Run time error

Answer: (a)

Explanation:

\_c=5+10=15.

In java variable name can be start with underscore (\_) and it can contain special character dollar singe ($)

(4)

public class Identifier {

public static void main(String[] args) {

int $=5;

final byte $$=-10;

int \_$=$+$$;

System.out.print(\_$);

}

}

What will output when you compile and run the above code?

(a)-5

(b) 5

(c) 105

(d) Error

Answer: (a)

Explanation:

-$=5+ (-10) =-5

In java programming language variable name includes alphabet, digits few special characters like underscore (\_), dollar singe ($).

(5)

public class Identifier {

public static void main(String[] args) {

int \_=5;

short \_\_ =2;

int \_\_\_=\_\_-~\_;

System.out.print(\_\_\_);

}

}

What will output when you compile and run the above code?

(a) 2

(b) 4

(c) 8

(d)Error

Answer: (c)

Explanation:

In java programming language variable name includes alphabet, digits few special characters like underscore (\_), dollar singe ($).

Here \_ (one underscore), \_\_ (two underscore) or \_\_\_ (three underscore) are valid name of any variable.

So, \_\_\_=2+~5=2-(-6) =2+6=8

Note: ~ is 1’s complementing operator

**abstract** **class** Abstract {

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

System.*out*.print("It is abstract class");

}

**static**{

System.*out*.println("Why");

}

}

What will output when you compile and run the above code?

(a) Why

It is **abstract** **class**

(b) It is **abstract** **class**

Why

(c)It is **abstract** **class**

(d)Compiler error

Answer: (a)

(2)

**abstract** **class** Abstract {

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

**double**[] array={1D,2d,3,4.,5.0};

display(array);

}

**abstract** **void** display();

**static** **void** display(**double**[] arr){

**for**(**double** d:arr)

System.out.println(d);

}

}

What will output when you compile and run the above code?

(a) 1.0

2.0

3.0

4.0

5.0

(b)1.0

(c)5.0

(d)Compiler error

Answer: (a)

(3)

**class** Abstract {

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

**float**[] array={1F,2f};

show(array);

}

**abstract** **void** display();

**static** **void** show(**float**[] arr){

**for**(**int** i=0;i<ARR.< span="">length;i++)

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

}

}

What will output when you compile and run the above code?

(a)1.0

2.0

(b)1.0

(c)1.0

2.0

null

(d)Compiler error

Answer: (d)

(4)

**abstract** **class** Abstract {

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

**if**(**new** Abstract() **instanceof** Abstract)

System.out.print("True");

**else**

System.out.print("False");

}

}

What will output when you compile and run the above code?

(a)True

(b)False

(c)Run time exception

(d)Compiler error

Answer: (d)

(5)

**abstract** **class** Abstract {

**static** **int** a=3;

**static** **float** b=5.3f;

Abstract(p,q){

a=p;

b=q;

}

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

System.out.print((**int**)a+~a+b);

}

}

What will output when you compile and run the above code?

(a)4.3

(b)4

(c)5

(d)Compiler error

Answer: (d)

abstract class can have a main method and can be run, only no object can be created for it. It can also have a constructor.

**abstract** **class** Test2{

Test2(**int** a){

}

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

**int** i=0;

**for**( i=-0;i<3;i++){

System.*out*.print(i);

} System.*out*.print(i);

}

}

**class** A **extends** Test2{

}

Compile error: Must define default constructor for superclass or provide same constructor in subclass. Also overridden constructor must use super(int a) as 1st line.

**interface** Apple{

**float** *cost*=9.5f;

**public** **void** display();

}

**class** Try **implements** Apple{

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

Try t=**new** Try();

t.display();

}

**public** **void** display(){

System.out.print("cost of apple is :"+cost);

}

}

What will output when you compile and run the above code?

(a) cost of apple is :9.5

(b) cost of apple is :0.0

(c) cost of apple is :9.5f

(d)Compiler error

Answer: (a)

(2)

**interface** Inf{

**double** *a*,*b*;

**public** **void** add(**double** p,**double** q);

}

**class** Try **implements** Inf{

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

Try t=**new** Try();

t.add(11.0,22.0);

}

**public** **void** add(**double** p,**double** q){

a=p;

b=q;

System.out.print(a+b);

}

}

What will output when you compile and run the above code?

(a)11.0

(b)22.0

(c)33.0

(d)Compiler error

Answer: (d)

(3)

**interface** Inf{

**public** **long** add(**int** p,**int** q);

**public** **void** display(**long** sum){

Systm.out.print(sum);

}

}

**class** Try **implements** Inf{

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

Try t=**new** Try();

**long** sum=t.add(130,120);

t.display(sum);

}

**public** **long** add(**int** p,**int** q){

**return** p+q;

}

}

What will output when you compile and run the above code?

(a)130

(b)120

(c)250

(d)Compiler error

Answer: (d) Cannot reduce the visibility of the overridden method.

(4)

**interface** Inf{

**long** cal(**int** p);

}

**class** Try **implements** Inf{

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

Try t=**new** Try();

**long** sum=t.cal(25);

System.out.print(sum);

}

**long** cal(**int** p){

**return** p++;

}

}

What will output when you compile and run the above code?

(a)25

(b)26

(c)27

(d)Compiler error

Answer: (d)

(5)

**interface** Inf{

**int** a=5;

}

**class** Test{

**int** b;

}

**class** Try **extends** Test **implements** Inf{

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

Try t=**new** Try();

System.out.print(a+t.b);

}

}

What will output when you compile and run the above code?

(a)5

(b)10

(c)Garbage value

(d)Compiler error

Answer: (a)

(6)

**class** Test{

**int** a=0;

**protected** **interface** Ni{

**short** calculation(**int** p);

}

}

**class** Try **extends** Test **implements** Test.Ni{

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

Try t=**new** Try();

**short** val=t.calculation(t.a);

System.out.print(val);

}

**public** **short** calculation(**int** p){

**for**(**int** i=0;i<2;i++){

**int** a=2;

++a;

}

**return** (**short**)(a+p);

}

}

What will output when you compile and run the above code?

(a)0

(b)3

(c)4

(d)Compiler error

Answer: (a)

(7)

**class** Mango{

**final** **int** a=5;

}

**class** Fruit **extends** Mango {

**final** **int** a=10;

}

**class** DynamicDispatch **extends** Fruit{

**final** **int** a=20;

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

Mango m=**new** DynamicDispatch();

Fruit f=**new** DynamicDispatch();

System.out.print(m.a|f.a);

}

}

What will be output of above program?

(a)15

(b)20

(c)30

(d)Compiler error

Answer: (a)

(8)

**class** Mango{

**final** **int** a=5;

}

**class** Fruit **extends** Mango {

**final** **int** a=10;

}

**class** DynamicDispatch **extends** Fruit{

**final** **int** a=15;

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

DynamicDispatch t=**new** Mango();

System.out.print(t.a);

}

}

What will be output of above program?

(a)5

(b)10

(c)15

(d)Compiler error

Answer: (d)

public class TypeConversion {

public static void main(String[] args) {

{

final int a='\15';

System.out.println(a);

}

int a=25;byte b=0100;

a=a+b;

System.out.println(a);

}

}

What will output when you compile and run the above code?

(a)15

125

(b)13

89

(c)Run time exception

(d) Compiler error

(2)

public class TypeConversion {

public static void main(String[] args) {

float f=12e-1F;

final long l=12L;

f=f+l;

System.out.println(f);

}

}

What will output when you compile and run the above code?

(a)18.9

(b)13.2

(c)Run time exception

(d) Compiler error

Answer: (b)

(3)

public class TypeConversion {

public static void main(String[] args) {

char a=5;

short b=-++a;

System.out.println(b);

}

}

What will output when you compile and run the above code?

(a)-4

(b)-5

(c)-6

(d) Compiler error

Answer: (d)

(4)

public class TypeConversion {

public static void main(String[] args) {

boolean b=11>=11;

String str=(String)b;

System.out.println(str);

}

}

What will output when you compile and run the above code?

(a)true

(b)false

(c)”true”

(d) Compiler error

Answer: (d)

(5)

public class TypeConversion {

public static void main(String[] args) {

byte b=016;

short s=0x16;

char c='\16';

int a=b+s+c;

System.out.println(a);

}

}

What will output when you compile and run the above code?

(a)48

(b)50

(c)Run time exception

(d) Compiler error

Answer: (b