01.

02. D- Compile time error

03. A-89

04.

05. Line 2, Line 3, Line 4 –The attribute variables length, width & height are declared on the class Box. But in the main method it is not called using the reference b1.

Output:

0

0

0

06. Prints the default values of the class Box that is referred by the reference variable b1.

Output:

0

0

0

07. Prints the assigned values of attributes of the class Box that is referred by the reference variable b1.

Output:

Default Constructor

2

2

2

08. We can create the default constructor explicitly and pass desired values to attributes of an instance except default values of their data types.

09. In this parameterized constructor we can assign values to attributes of the instance directly through the constructor when we are creating an instance

Output:

Volume : 60

Volume : 180

10. Line 4, Line 5, Line 6……You cannot assign values to attributes on another line after declaration. If you want to,

you must use an instance block, else assign values at the moment of the declaration of the attributes.

11. The main purpose of a constructor is to initialize the **instance variables of a class**. Constructor is a block of

code that initializes the newly created object. A constructor resembles an instance method in java but it’s not a

method as it doesn’t have a return type. In short constructor and method are different. People often refer

constructor as special type of method in Java.

|  |  |
| --- | --- |
| **Java Constructor** | **Java Method** |
| A constructor is used to initialize the state of an object. | A method is used to expose the behavior of an object. |
| A constructor must not have a return type. | A method must have a return type. |
| The constructor is invoked implicitly. | The method is invoked explicitly. |
| The Java compiler provides a default constructor if you don't have any constructor in a class. | The method is not provided by the compiler in any case. |
| The constructor name must be same as the class name. | The method name may or may not be same as the class name. |

12.

13. A. Box b1;

B. Box b1 = new Box();

C. new Box;

14. Compile Error occurs.

Line 2 – There is no attribute declared as “WIDTH”. But there is “width”. Hence as variables are case sensitive they are two different variables.

Line 3 – There is no attribute variable declared as “height”.

15. C - 1234

16. A. int x;

int y;

MyClass(int i, int j){ x=i; y=j; }

17.

Output:

100 101

0 0

After the address are equalized(a1=a2), a1 also referring the address of a2. And the instance a1 is removed from the heap by garbage collector.

Output:

100 101

0 0

18.

Output:

100 101

0 0

19.

20. B. void A(int i){a=i;}

C. A(){}

D. Insert nothing

21.

Line 1 -

Line 2 -

Line 3 -

Line 4 -

Line 5 -

Line 6 -

Line 7 -

Line 8 -

Line 9 -

Line 10 -

Line 11 -

Line 12 -

22. A. A(int i){a=i;}

D. A(int a){this.a=a;}

E. A(int i){a=100;}

23. D. Compile Error at line 2.

24.

25.

26.

class Date{

private int year =1970;

private int month = 1;

private int day = 1;

public void printDate(){

System.out.println(year+"-"+month+"-"+day);

}

public void setYear(int year){

this.year=year;

}

public void setMonth(int month){

this.month=month;

}

public void setDay(int day){

this.day=day;

}

public int getYear(){

return this.year;

}

public int getMonth(){

return this.month;

}

public int getDay(){

return this.day;

}

}

class Demo{

public static void main(String args[]){

Date d1 = new Date();

d1.printDate(); //1970-1-1

d1.setYear(2016);

d1.setMonth(5);

d1.setDay(31);

System.out.println("Year : "+d1.getYear());

System.out.println("Month : "+d1.getMonth());

System.out.println("Day : "+d1.getDay());

}

}

27. A. Compile error at 1

28. F. None of above

29.

30. Line 3, Line 4, Line 7, Line 8

31.

32.

33.

34.

35.

class Date{

static int DAY=1;

static int MONTH=2;

static int YEAR=3;

private int year =1970;

private int month = 1;

private int day = 1;

public void printDate(){

System.out.println(year+"-"+month+"-"+day);

}

public void setYear(int year){

this.year=year;

}

public void setMonth(int month){

this.month=month;

}

public void setDay(int day){

this.day=day;

}

public int getYear(){

return this.year;

}

public int getMonth(){

return this.month;

}

public int getDay(){

return this.day;

}

public void set(int field,int value){

if (field==DAY){

this.day=value;

}else if (field==MONTH){

this.month=value;

}else{

this.year=value;

}

}

}

class Demo{

public static void main(String args[]){

Date d1 = new Date();

d1.printDate(); //1970-1-1

d1.setYear(2016);

d1.setMonth(5);

d1.setDay(31);

System.out.println("Year : "+d1.getYear());

System.out.println("Month : "+d1.getMonth());

System.out.println("Day : "+d1.getDay());

d1.printDate();

d1.set(Date.YEAR,2016); //set(int field, int value)

d1.set(Date.MONTH,05);

d1.set(Date.DAY,30);

d1.printDate();

}

}

56.

import java.util.\*;

class Stack{

int currentPos;

static int[] array=new int[100];

public void push(int push){

array[currentPos]=push;

currentPos++;

}

public void printStack(){

int[] arrayTemp=new int[currentPos];

for (int i = 0,j=arrayTemp.length-1; i<arrayTemp.length ; i++,j--){

arrayTemp[i]=array[j];

}

System.out.println(Arrays.toString(arrayTemp));

}

public int size(){

return currentPos;

}

public void pop(){

currentPos--;

}

Stack(){

}

}

class Demo{

public static void main(String args[]){

Stack s1 = new Stack();

s1.push(60);

s1.push(40);

s1.push(30);

s1.push(50);

s1.push(20);

s1.printStack(); //[20,50,30,40,60]

System.out.println("Size of the stack is : "+s1.size()); //prints 5

s1.pop(); //Remove an item of the stack

s1.printStack(); //[50,30,40,60]

System.out.println("Size of the stack is : "+s1.size()); //prints 4

}

}