Q. In Java, each thread has its own \_\_\_\_\_\_\_\_, in which it runs ?

|  |  |
| --- | --- |
| **A.** main() method | **B.** JVM |
| **C.** Call stack | **D.** Memory |

**Correct Answer : OPTION C, call stack. Thread of execution means an individual 'lightweight' process that has its own call stack. In java each thread has its own call stack.**

Q. In Java, by default every thread is given a \_\_\_\_\_\_\_\_\_ .

|  |  |
| --- | --- |
| **A.** MIN\_PRIORITY(0) | **B.** NORM\_PRIORITY(5) |
| **C.** MAX\_PRIORITY(10) | **D.** Any Random Priority |

**Correct Answer : OPTION B, NORM\_PRIORITY(5)**

Q. What will happen if we call run() directly, without start() ?

|  |
| --- |
| **A.** Program will give a compilation error. |
| **B.** Nothing will happen both the methods are same. |
| **C.** Runtime error. |
| **D.** thread won't be allocated a new call stack, and start running in the current call stack. |

**Correct Answer : OPTION D, thread won't be allocated a new call stack, and start running in the current call stack**

Q. Which two are valid constructors for Thread class ?

1. Thread(Runnable r, String name)
2. Thread()
3. Thread(int priority)
4. Thread(Runnable r, ThreadGroup g)

|  |  |
| --- | --- |
| **A.** 1 and 2 | **B.** 1 and 3 |
| **C.** 2 and 3 | **D.** 2 and 4 |

**Correct Answer : OPTION A, 1 and 2**

Q. What Exception is thrown when you start a thread twice ?

|  |  |
| --- | --- |
| **A.** InterruptedException | **B.** NullPointerException |
| **C.** IOException | **D.** IllegalStateException |

**Correct Answer : OPTION D, IllegalStateException**

Q. Which class or interface defines the wait(), notify(), and notifyAll() methods ?

|  |  |
| --- | --- |
| **A.** Object | **B.** Thread |
| **C.** Runnable | **D.** Class |

**Correct Answer : OPTION A, Object. Java provides benefit of avoiding thread pooling using interthread communication. The wait(), notify(), notifyAll() are of Object class. These methods are implemented as final in Object. All three methods can be called only from within a synchronized context.**

Q. Which method will contain the body of the thread ?

|  |  |
| --- | --- |
| **A.** run() | **B.** start() |
| **C.** stop() | **D.** main() |

**Correct Answer : OPTION A, run()**

Q. What is the Output of given code ?

class MyThread extends Thread

{

public static void main(String[] args)

{

MyThread my = new MyThread();

Thread t = new Thread(my);

t.start();

}

public void run()

{

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

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

} } }

|  |  |
| --- | --- |
| **A.** compilation error | **B.** ..0..1..2.. |
| **C.** 0..2..2. | **D.** 0..1..2.. |

**Correct Answer : OPTION D, 0..1..2.. (The thread MyThread will start and loop three times.)**

Q. Which of the following statement is not correct ?

|  |
| --- |
| **A.** notify() method wakes up a thread that called wait() on same object. |
| **B.** sleep() is a static method simply used to put your thread on sleep. |
| **C.** You can override wait() method to give your own implemention. |
| **D.** The notifyAll() method must be called from a synchronised context. |

**Correct Answer : OPTION C, You can override wait() method to give your own implemention.**

Q. What will be the Output of given code ?

public class Test implements Runnable

{

public void run()

{

System.out.println("r1 ");

System.out.println("r2 ");

}

public static void main( String[] args )

{

Thread t = new Thread(new Test());

t.start();

System.out.println("m1 ");

t.join();

System.out.println("m2 ");

}

}

|  |  |
| --- | --- |
| **A.** compilation error. | **B.** r1 r2 m1 m2 |
| **C.** m1 m2 r1 r2 | **D.** m1 r1 r2 m2 |

**Correct Answer : OPTION A, Compilation error. The join() method must be placed in a try -catch block.**

Q. Give One Word for ?

A situation where two or more threads are blocked forever and waiting for each other to release resources.

|  |  |
| --- | --- |
| **A.** Critical Section | **B.** Mutual Exclusion |
| **C.** Deadlock | **D.** Synchronization |

**Correct Answer : OPTION C, Deadlock**

Q. Which of the following is not a primitive data types?

|  |  |
| --- | --- |
| **A.** Byte. | **B.** String. |
| **C.** Integer. | **D.** Float. |

Q. What is the range of the char type?

|  |  |
| --- | --- |
| **A.** 0 to 2^16. | **B.** 0 to 2^15. |
| **C.** 0 to 2^16-1. | **D.** 0 to 2^15-1. |

Q. What is the output of this program?

class bitwise\_operator {

public static void main(String args[])

{

int var1 = 42;

int var2 = ~var1;

System.out.print(var1 + " " + var2);

}

}

|  |
| --- |
| **A.** 42 43 |
| **B.** 42 -43 |
| **C.** 42 42 |
| **D.** 42 -42 |

Q. Object is an \_\_\_\_\_\_\_\_\_ of a class

|  |  |
| --- | --- |
| **A.** Instance. | **B.** Implement. |
| **C.** Inheritance. | **D.** Invoke. |

Q. Class is a \_\_\_\_\_\_ entity.

|  |  |
| --- | --- |
| **A.** Logical. | **B.** Physical. |
| **C.** Up normal. | **D.** Collection of. |

Q. Object is a \_\_\_\_\_\_\_ entity.

|  |  |  |  |
| --- | --- | --- | --- |
| **A.** Normal | **B.** Physical | **C.** Logical | **D.** Collection of |

Q. \_\_\_\_\_\_ can appear anywhere in the body of a java method.

|  |  |
| --- | --- |
| **A.** Definition. | **B.** Declaration. |
| **C.** Determine. | **D.** None of the above. |

Q. \_\_\_\_\_\_ must be the first non comment statement in the file.

|  |  |
| --- | --- |
| **A.** Package. | **B.** Class. |
| **C.** Object. | **D.** Declaration. |

Q. \_\_\_\_\_\_\_ is passed to a method by use of call-by-reference.

|  |  |
| --- | --- |
| **A.** Variables. | **B.** Objects. |
| **C.** Methods. | **D.** Operators. |

Q. Every method of a \_\_\_\_\_\_\_\_ is implicitly final.

|  |  |
| --- | --- |
| **A.** Static class. | **B.** Dynamic class. |
| **C.** Final class. | **D.** Abstract class. |

Q. A \_\_\_\_\_\_\_ object cannot be modified after it is created.

|  |  |
| --- | --- |
| **A.** Double. | **B.** Int. |
| **C.** String. | **D.** Main. |

Q. A \_\_\_\_ class may not have any abstract method.

|  |  |
| --- | --- |
| **A.** Abstract. | **B.** Static. |
| **C.** Final. | **D.** Public. |

Q. What will be the result of the expression 13 & 25?

|  |  |
| --- | --- |
| **A.** 38 | **B.** 25 |
| **C.** 9 | **D.** 12 |

Q. What will be result of expression 9 | 9?

|  |  |
| --- | --- |
| **A.** 1 | **B.** 18 |
| **C.** 9 | **D.** 12 |

Q. \_\_\_\_\_\_\_\_\_ Operators are overloaded for string objects?

|  |  |
| --- | --- |
| **A.** - , + | **B.** + , = |
| **C.** << , >> | **D.** ++ , -- |

Q. Which of these cannot be declared static?

|  |  |
| --- | --- |
| **A.** class | **B.** variable |
| **C.** instance | **D.** method |

**Correct Answer : OPTION C, static statements are executed as soon as class containing them is loaded and prior to any object declaration. Thus, instance, which is created after object declaration, cannot be static.**

Q. What is the output of this program?

class **access**{

public int x;

static int y;

void cal(int a, int b){

x += a ;

y += b;

}

}

class **static\_specifier** {

public static void main(String args[])

{

access obj1 = new access();

access obj2 = new access();

obj1.x = 0;

obj1.y = 0;

obj1.cal(1, 2);

obj2.x = 0;

obj2.cal(2, 3);

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

}

}

|  |  |
| --- | --- |
| **A.** 1 2 | **B.** 1 5 |
| **C.** 2 3 | **D.** 2 5 |

**Correct Answer : OPTION B**

Q. A package is a collection of \_\_\_\_\_\_.

|  |
| --- |
| **A.** Keywords. |
| **B.** Directory structure, Classes and Interfaces. |
| **C.** Editing tools. |
| **D.** Views. |

**Correct Answer : OPTION B**

Q. \_\_\_\_\_\_\_\_ statement is valid for array declaration.

|  |  |
| --- | --- |
| **A.** Int number(); | **B.** Float number(); |
| **C.** Float number[]; | **D.** Count Int[]; |

**Correct Answer : OPTION C**

Q. What is the output of relational operators?

|  |  |
| --- | --- |
| **A.** Integer. | **B.** Boolean. |
| **C.** Character. | **D.** Double. |

**Correct Answer : OPTION B**

Q. Which of these operators can skip evaluating right hand operand?

|  |  |  |  |
| --- | --- | --- | --- |
| **A.** ! | **B.** | | **C.** & | **D.** && |

**Correct Answer : OPTION D**

Q. Which exception is thrown by the read() method of Input Stream class?

|  |  |
| --- | --- |
| **A.** Exception. | **B.** ClassNotFoundException. |
| **C.** ReadException. | **D.** IOException. |

**Correct Answer : OPTION D**

Q. Which of the following denotes a javadoc comment?

|  |  |
| --- | --- |
| **A.** //# | **B.** /\* |
| **C.** /\*\* | **D.** //\*\* |

**Correct Answer : OPTION C, while /\* \*/ is used for code comment, javadoc comment is placed inside /\*\* \*\*/**

Q. Using the keyword Interface you can fully abstract a \_\_\_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| **A.** Method | **B.** Keyword |
| **C.** Class | **D.** Variables |

**Correct Answer : OPTION C**

Q. One interface can inherit another by use of the keyword \_\_\_\_\_\_?

|  |
| --- |
| **A.** Public |
| **B.** Extends |
| **C.** Implements |
| **D.** Inherits |

**Correct Answer : OPTION B**

Q. An exception is an \_\_\_\_\_\_\_\_\_ condition that arises in a code.

|  |  |
| --- | --- |
| **A.** Abnormal | **B.** Casual |
| **C.** Unfortunate | **D.** Opposite |

**Correct Answer : OPTION A**

Q. \_\_\_\_\_\_\_ is at the top of the exception class hierarchy.

|  |  |
| --- | --- |
| **A.** Try | **B.** Throwable |
| **C.** Exception Class | **D.** Catch |

**Correct Answer : OPTION B**

Q. In java, thread to thread communication is called \_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| **A.** Passing | **B.** Sending |
| **C.** Messaging | **D.** Calling |

**Correct Answer : OPTION C**

Q. Which of these access specifiers can be used for an Interface?

|  |
| --- |
| **A.** Public |
| **B.** Protected |
| **C.** Private |
| **D.** All of the above |

**Correct Answer : OPTION A**

Q. Java programs perform I/O through \_\_\_\_\_\_\_\_.

|  |  |
| --- | --- |
| **A.** I/O methods | **B.** I/O package |
| **C.** Streams | **D.** Compiler |

**Correct Answer : OPTION C**

Q. In Java a \_\_\_\_\_\_\_ is a sequence of characters.

|  |  |
| --- | --- |
| **A.** String | **B.** ArrayChar |
| **C.** GroupChar | **D.** Collection |

**Correct Answer : OPTION A**

Q. The String is defined in \_\_\_\_\_\_\_ namespace.

|  |  |
| --- | --- |
| **A.** Java.lang | **B.** Java.string |
| **C.** Java.char | **D.** Java.awt |

**Correct Answer : OPTION A**

Q. \_\_\_\_\_\_\_\_ is a special member function.

|  |  |
| --- | --- |
| **A.** Method | **B.** Class |
| **C.** User defined function | **D.** Constructor |

**Correct Answer : OPTION D**

Q. Keyword \_\_\_\_\_\_\_ is always a reference to the object.

|  |  |
| --- | --- |
| **A.** New | **B.** This |
| **C.** Invoke | **D.** Class |

**Correct Answer : OPTION B**

Q. \_\_\_\_\_\_\_\_ is the mechanism that binds together the code and the data.

|  |  |
| --- | --- |
| **A.** Polymorphism. | **B.** Encapsulation. |
| **C.** Inheritance. | **D.** Together. |

**Correct Answer : OPTION B**

Q. Java is designed for \_\_\_\_\_\_\_ environment of the internet.

|  |  |  |  |
| --- | --- | --- | --- |
| **A.** Development | **B.** Deducting | **C.** Distributed | **D.** Web design |

**Correct Answer : OPTION C**

Q. \_\_\_\_\_\_\_ is a small unit of a process.

|  |  |
| --- | --- |
| **A.** Method | **B.** Thread |
| **C.** Applet | **D.** Stream |

**Correct Answer : OPTION B**

Q. Wrapper class is a wrapper around a \_\_\_\_\_\_\_ data type.

|  |  |
| --- | --- |
| **A.** Normal | **B.** Central |
| **C.** Primitive | **D.** Concrete |

**Correct Answer : OPTION C**

Q. Once an interface has been defined, one or more \_\_\_\_\_\_\_ can implement that interface.

|  |  |
| --- | --- |
| **A.** Class | **B.** Object |
| **C.** Methods | **D.** Keywords |

**Correct Answer : OPTION A**

Q. Variable declared as \_\_\_\_\_\_\_\_ do not occupy memory on a per instance basis.

|  |  |
| --- | --- |
| **A.** Static | **B.** Final |
| **C.** Abstract | **D.** Code |

**Correct Answer : OPTION B**

Q. \_\_\_\_\_\_\_ must be the first statement executed inside a subclass Constructor.

|  |  |
| --- | --- |
| **A.** final() | **B.** super() |
| **C.** static{} | **D.** None of these |

**Correct Answer : OPTION B**

Q. Which function is used to perform some action when the object is to be destroyed?

|  |  |
| --- | --- |
| **A.** finalize() | **B.** delete() |
| **C.** main() | **D.** None of the above |

**Correct Answer : OPTION A**

Q. The \_\_\_\_\_\_\_\_ statement is used to terminate a statement sequence.

|  |  |
| --- | --- |
| **A.**Break | **B.**Switch |
| **C.** Continue | **D.** Wait |

**Correct Answer : OPTION A**

Q. Multidimensional arrays are actually \_\_\_\_\_\_\_\_.

|  |
| --- |
| **A.** Arrays of element. |
| **B.**Array and variable. |
| **C.** Arrays of arrays. |
| **D.** None of the above |

**Correct Answer : OPTION C**

Q. \_\_\_\_\_\_\_ statement in java is multiway branch statement.

|  |  |
| --- | --- |
| **A.** Declaration | **B.** Case |
| **C.** Multi | **D.** Switch |

**Correct Answer : OPTION D**

Q. Which class cannot be a subclass in java?

|  |  |
| --- | --- |
| **A.** Abstract class | **B.** Parent class |
| **C.** Final class | **D.** None of the above |

**Correct Answer : OPTION C, Final Class**

Q. A Suspended thread can be revived by using?

|  |  |
| --- | --- |
| **A.** start() method | **B.** Suspend() method |
| **C.** resume() method | **D.** yield() method |

**Correct Answer : OPTION C, resume() method**

Q. Which collection class associates values with keys, and orders the keys according to their natural order?

|  |  |
| --- | --- |
| **A.** java.util.HashSet | **B.** java.util.LinkedList |
| **C.** java.util.TreeMap | **D.** java.util.SortedSet |

**Correct Answer : OPTION C, java.util.TreeMap**

Q. Which methods are utilized to control the access to an object in multi-threaded programming?

|  |  |
| --- | --- |
| **A.** Asynchronized methods | **B.** Synchronized methods |
| **C.** Serialized methods | **D.** None of above |

**Correct Answer : OPTION B, Synchronized methods**

Q. In Runnable, many threads share the same object instance. True or False?

|  |  |
| --- | --- |
| **A.** True | **B.** False |

**Correct Answer : OPTION A, True**

Q. Java beans have no types. True or False?

|  |  |
| --- | --- |
| **A.** True | **B.** False |

**Correct Answer : OPTION A, True**

Q. What is the output of this program?

class average {

public static void main(String args[])

{

double num[] = {5.5, 10.1, 11, 12.8, 56.9, 2.5};

double result;

result = 0;

for (int i = 0; i < 6; ++i)

result = result + num[i];

System.out.print(result/6);

}

}

|  |  |
| --- | --- |
| **A.** 16.34 | **B.** 16.5555 |
| **C.** 16.46666666666667 | **D.** 16.4666666666 |

**Correct Answer : OPTION C, 16.46666666666667. Simple Average is performed over all the numbers.**

Q. What is the output of this program?

class conversion {

public static void main(String args[])

{

double a = 295.04;

int b = 300;

byte c = (byte) a;

byte d = (byte) b;

System.out.println(c + " " + d);

}

}

|  |  |
| --- | --- |
| **A.** 38 43 | **B.** 39 44 |
| **C.** 295 300 | **D.** 295.4 300.6 |

**Correct Answer : OPTION B, 39 44. Type casting a larger variable into a smaller variable results in modulo of larger variable by range of smaller variable. b contains 300 which is larger than byte’s range i:e -128 to 127 hence d contains 300 modulo 256 i:e 44.**

Q. What is the output of this program?

class increment {

public static void main(String args[])

{

int g = 3;

System.out.print(++g \* 8);

}

}

|  |  |
| --- | --- |
| **A.** 25 | **B.** 24 |
| **C.** 32 | **D.** 33 |

**Correct Answer : OPTION C, 32. Operator ++ has more preference than \*, thus g becomes 4 and when multiplied by 8 gives 32.**

Q. Which of these operators is used to allocate memory to array variable in Java?

|  |  |
| --- | --- |
| **A.** malloc | **B.** new malloc |
| **C.** new | **D.** calloc |

**Correct Answer : OPTION C, new. Operator new allocates block of memory specified by the size of array, and gives the reference of memory allocated to the array variable.**

Q. What is the output of this program?

class array\_output {

public static void main(String args[])

{

int array\_variable [] = new int[10];

for (int i = 0; i < 10; ++i) {

array\_variable[i] = i;

System.out.print(array\_variable[i] + " ");

i++;

}

}

}

|  |  |
| --- | --- |
| **A.** 0 2 4 6 8 | **B.** 1 3 5 7 9 |
| **C.** 0 1 2 3 4 5 6 7 8 9 | **D.** 1 2 3 4 5 6 7 8 9 10 |

**Correct Answer : OPTION A, 0 2 4 6 8. When an array is declared using new operator then all of its elements are initialized to 0 automatically. for loop body is executed 5 times as whenever controls comes in the loop i value is incremented twice, first by i++ in body of loop then by ++i in increment condition of for loop.**

Q. Which statement is static and synchronized in JDBC API?

|  |  |
| --- | --- |
| **A.** executeQuery() | **B.** executeUpdate() |
| **C.** getConnection() | **D.** prepareCall() |

**Correct Answer : OPTION C, getConnection()**

Q. All raw data types should be read and uploaded to the database as an array of?

|  |  |
| --- | --- |
| **A.** int | **B.** char |
| **C.** boolean | **D.** byte |

**Correct Answer : OPTION D, byte**

Q. The class java.sql.Timestamp is associated with?

|  |  |
| --- | --- |
| **A.** java.util.Time | **B.** java.sql.Time |
| **C.** java.util.Date | **D.** None of the above |

**Correct Answer : OPTION C, JAVA.Util.Date**

Q. Which method executes only once?

|  |  |
| --- | --- |
| **A.** start() | **B.** stop() |
| **C.** init() | **D.** destroy() |

**Correct Answer : OPTION C, init()**

Q. Which of the following is a legal ideantifier in java ?

|  |  |
| --- | --- |
| **A.** 2variable | **B.** #myvar |
| **C.** +@$var | **D.** $\_myvar |

**Correct Answer : OPTION D, $\_myvar. Because all identifier must start with either a letter( a to z or A to Z ) or currency character($) or an underscore. After the first character,an identifier can have any combination of character.**

Q. Which of these is NOT a valid keyword or reserved word in Java ?

|  |  |
| --- | --- |
| **A.** default | **B.** null |
| **C.** String | **D.** volatile |

**Correct Answer : OPTION C, String is not a keyword or a reserved word in java**

Q. Which is the legal range of values for a short ?

|  |  |
| --- | --- |
| **A.** -128 to 127 | **B.** -256 to 255 |
| **C.** -32768 to 32767 | **D.** 0 to 65535 |

**Correct Answer : OPTION C, Legal range for short data type is between -32768 to 32767.**

Q. Which of the following Array declaration statement is illegal ?

|  |  |
| --- | --- |
| **A.** int [ ] a [ ] = new int [4] [4]; | **B.** int a[ ][ ] = new int [4] [4]; |
| **C.** int a[ ] [ ] = new int [ ] [4]; | **D.** int [ ] a [ ] = new int [4] [ ]; |

**Correct Answer : OPTION C, int a[ ] [ ] = new int [ ] [4]; because the size of the rows must always be mentioned.**

Q. What will happen if you try to compile and run the following code ?

public class Test

{

public static void main(String argv[])

{

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

System.out.println(arr[1]);

}

}

|  |  |
| --- | --- |
| **A.** 1 | **B.** Error: arr is referenced before it is initialized |
| **C.** 2 | **D.** Error: size of array must be defined |

**Correct Answer : OPTION C, 2**

Q. In the given Array declaration, which expression returns the output as 5 ?

int [] arr = { 23, 5, 78, 34, 2};

|  |  |
| --- | --- |
| **A.** arr.length() | **B.** arr[].length() |
| **C.** arr[].length | **D.** arr.length |

**Correct Answer : OPTION D, arr.length will print the size of array arr which is 5. length is not a method it is a operator used in java that return size of an array.**

Q. What will be the output upon the execution of the following code ?

public class Test

{

public static void main(String[] args)

{

int j = 5;

for (int i = 0; i< j; i++)

{

if ( i <= j-- )

System.out.print( (i\*j) + " ");

}

}

}

|  |  |
| --- | --- |
| **A.** 0 3 2 | **B.** 1 2 3 |
| **C.** 0 3 4 | **D.** 1 4 2 |

**Correct Answer : OPTION C, 0 3 4. Here for-loop will execute only three times. In first pass the value of i is 0 and j is 4. So 0\*4 = 0 is printed. In second pass the value of i is 1 and 3 so it will print 3. In last pass the value of i is 2 and 2 so it will print 4**

Q. Which declaration of the main() method is valid ?

|  |  |
| --- | --- |
| **A.** public void main( String args [] ) | **B.** public void static main( String args [] ) |
| **C.** final public static void main(String[] arr) | **D.** public static void main( String[] arr) |

**Correct Answer : OPTION D, public static void main( String []arr)**

Q. Is 3 \* 4 equivalent to 3 << 2 ?

|  |  |
| --- | --- |
| **A.** Yes | **B.** No |

**Correct Answer : OPTION A, Yes**

Q. Which one of the following is invalid declaration of a char ?

|  |  |
| --- | --- |
| **A.** char c1 = 064770; | **B.** char c2 = 'face'; |
| **C.** char c3 = 0xbeef; | **D.** char c4 = '\uface'; |

**Correct Answer : OPTION B, char c2 = 'face'**

Q. What will be the output of the following code ?

public class Test {

public static void main(String[] args)

{

double d = 100.04;

float f = d;

System.out.println("Float value "+f);

}

}

|  |  |
| --- | --- |
| **A.** 100.04 | **B.** 100.0 |
| **C.** Compilation Error (lossy conversion from double to float) | **D.** 100 |

**Correct Answer : OPTION C, Compilation error. Because of incompatible type, assigning a larger type value to a variable of smaller type value.**

Q. Is it possible in Java to create arrays of length zero ?

|  |
| --- |
| **A.** Yes, you can create arrays of any type with length zero. |
| **B.** No, it is not possible to create arrays of length zero in Java. |

**Correct Answer : OPTION A. Yes, we can create arrays of any type with length zero but we can not do anything with it.**

Q. Which operator will always evaluate all the Operands?

|  |  |
| --- | --- |
| **A.** || | **B.** && |
| **C.** ?: | **D.** % |

**Correct Answer : OPTION D, %(modulus) will always evaluate both its operand. || will not evaluate the second condition if the first passes. && will not evaluate the second condition if the first fails. And ?: is also conditional.**

Q. Which of the following is not a valid declaration of a Top level class ?

|  |  |
| --- | --- |
| **A.** final public class Test {} | **B.** class $Test{} |
| **C.** static class Test {} | **D.** public abstract class \_Test {} |

**Correct Answer : OPTION C, because a class can have only public or default(no modifier) access specifier. It can be either abstract, final or concrete (normal class). Only Inner classes in Java can be static, not Top level class.**

Q. How can such a restriction be enforced ?

A method within a class is only accessible by classes that are defined within the same package as the class of the method.

|  |  |
| --- | --- |
| **A.** Declare the method with the keyword public | **B.** Declare the method with keyword protected |
| **C.** Declare the method with keyword private | **D.** Without any accessibility specifiers. |

**Correct Answer : OPTION D, Do not declare the method with any accessibility specifiers. Because default modifier has scope only inside package.**

Q. A special method that is used to initialize a class object ?

|  |  |
| --- | --- |
| **A.** abstract method | **B.** static method |
| **C.** Constructor | **D.** overloaded method. |

**Correct Answer : OPTION C. Constructor is a special method that is used to initialize an object. Every class has a constructor, if we don't explicitly declare a constructor for any Java class the compiler builds a default constructor for that class.**

Q. Two methods are said to be overloaded if they have,

|  |
| --- |
| **A.** same name and same number of parameter but different return type. |
| **B.** they have same name. |
| **C.** they have different name but same number of argument. |
| **D.** have same name but different parameters. |

**Correct Answer : OPTION D, If two or more method in a class have same name but different parameters, it is known as Method Overloading**

Q. What will be the output of the program ?

class A

{

int x = 10;

public void assign(int x)

{

x = x;

System.out.println(this.x);

}

public static void main(String[] args)

{

new A().assign(100);

}

}

|  |  |
| --- | --- |
| **A.** 10 | **B.** 100 |
| **C.** 0 | **D.** compile-time error |

**Correct Answer : OPTION A, In the statement x = x; x is a local variable declared inside parameter of assign() method. And as you know local variable will always get preference over instance variable. So local variable x will assign the value 100 to itself. And when we print this.x, it will print the value of instance variable x which is 10.**

Q. Which statement does not create an object of class Student{} ?

|  |
| --- |
| **A.** new Student(); |
| **B.** Student s1 = new Student(), s2 = new Student(); |
| **C.** Student s1; |
| **D.** Student ss = new Student(); |

**Correct Answer : OPTION C, The statement Student s1 only declares a reference variable s1 of Student class not an object**

Q. this keyword in java is used to ?

|  |  |
| --- | --- |
| **A.** refer to current class object. | **B.** refer to static method of the class. |
| **C.** refer to parent class object. | **D.** refer to static variable of the class. |

**Correct Answer : OPTION A, this keyword refer to current class object.**

Q. What is the prototype of the default constructor for given class?

public class Test { }

|  |  |
| --- | --- |
| **A.** Test( ) | **B.** public Test( ) |
| **C.** Test(void) | **D.** public Test(void) |

**Correct Answer : OPTION B, public Test( ). The default constructor has the same access modifier as the class.**

Q. Which method is called by Garbage collection thread just before collecting eligible Objects ?

|  |  |
| --- | --- |
| **A.** finally() | **B.** finalize() |
| **C.** final() | **D.** gc() |

**Correct Answer : OPTION B, finalize() method is called by garbage collection thread before collecting object. Its the last chance for any object to perform cleanup utility.**

Q. Garbage Collection in java is done by who?

|  |  |
| --- | --- |
| **A.** Java Compiler | **B.** Object class |
| **C.** JVM | **D.** System class |

**Correct Answer : OPTION C, In Java destruction of object from memory is done automatically by the JVM. When there is no reference to an object, then that object is assumed to be no longer needed and the memory occupied by the object are released. This technique is called Garbage Collection. This is accomplished by the JVM.**

Q. What will be the output of the following program?

class B

{

static int count = 100;

public void increment()

{

count++;

}

public static void main(String []args)

{

B b1 = new B();

b1.increment();

B b2 = new B();

System.out.println(b2.count); // line 13

}

}

|  |  |
| --- | --- |
| **A.** 100 | **B.** 101 |
| **C.** Error in line 13 | **D.** 0 |

**Correct Answer : OPTION B, 101. Static variable has only one single storage. All the objects of the class having static variable will have the same instance of static variable.**

Q. Which of the following statement declares a constant field in Java?

|  |  |
| --- | --- |
| **A.** const int x = 10; | **B.** static int x = 10; |
| **C.** final static int x = 10; | **D.** volatile int x =10; |

**Correct Answer : OPTION C, final static int x = 10; because, Final modifier is used to declare a field as final i.e. it prevents its content from being modified. Final field must be initialized when it is declared.**

Q. Given the following code, which line will generate an error ?

class Test

{

static int x = 100; // line 3

int y = 200; // line 4

public static void main(String []args)

{

final int z; // line 7

z = x + y; // line 8

System.out.println(z);

}

}

|  |  |
| --- | --- |
| **A.** line 3 | **B.** line 4 |
| **C.** line 7 | **D.** line 8 |

**Correct Answer : OPTION D, line 8, because non-static variable y cannot be referenced from a static context i.e main method.**

Q. What will happen if you try to compile and run the following code ?

class Test

{

int x;

Test(int n)

{

System.out.println(x=n); // line 6

}

public static void main(String []args)

{

Test n = new Test(); // line 10

}

}

|  |  |
| --- | --- |
| **A.** Program exits without printing anything | **B.** Compilation error at line 10 |
| **C.** Compilation error at line 6 | **D.** Run-time exception |

**Correct Answer : OPTION B, Compilation error at line 10. Java compiler will throw an error becuase it does not find default constructor in class Test. As class Test has a parameterised constructor. So default constructor will not be autoamtically created.**

Q. Can you make a Constructor final ?

|  |  |
| --- | --- |
| **A.** Yes | **B.** No |

**Correct Answer : OPTION B, No. Although a constructor cannot be overriden but you cannot make a constructor as final.**

Q. Which of the given statement is not true about a Java Package?

|  |
| --- |
| **A.** A package can be defined as a group of similar types of classes and interface. |
| **B.** Package are used in order to avoid name conflicts and to control access of classes and interface. |
| **C.** A package cannot not have another package inside it. |
| **D.** Java uses file system directory to store package |

**Correct Answer : OPTION C, A package cannot not have another package inside it. This is wrong.**

Q. Given two files, what will be the Output?

**package pck1;**

public class A

{

int x = 10;

public int y = 20;

}

**package pck2;**

import pck1.\*;

class Test

{

public static void main(String []args)

{

A a = new A(); // line 7

System.out.println(a.x); // line 8

System.out.println(a.y); // line 9

}

}

|  |
| --- |
| **A.** 10 20 |
| **B.** compilation error at line 7 |
| **C.** compilation error at line 8 |
| **D.** compilation error at line 9 |

**Correct Answer : OPTION C, compilation error at line 8. Variable x has a default acccess, so it cannot be accessed from outside the package.**

Q. import keyword is used to?

|  |
| --- |
| **A.** both built-in packages and user-defined packages into your java source file. |
| **B.** import only built-in packages into your java source file |
| **C.** import only user-defined packages into your java source file |
| **D.** None of the above |

**Correct Answer : OPTION A, import both built-in packages and user-defined packages into your java source file.**

Q. You can import only static members of a class present in some other package using \_\_\_\_\_\_\_\_\_\_?

|  |  |
| --- | --- |
| **A.** import keyword | **B.** import static keyword |
| **C.** package keyword | **D.** static import keyword |

**Correct Answer : OPTION B, import static keyword. import static package.class-name.static-member-name this is how a static member is imported to a java source file.**

Q. Which of the statement is false about an abstract class?

|  |
| --- |
| **A.** An abstract class may not contain abstract methods. |
| **B.** An abstract class cannot have normal method. |
| **C.** An abstract class cannot be instantiated. |
| **D.** An abstract class can be extended. |

**Correct Answer : OPTION B, An abstract class cannot have normal method. This is wrong because If a class contain any abstract method then the class is declared as abstract class, but it can still have normal methods too. An abstract class is never instantiated. It is used to provide abstraction.**

Q. Fill in the blank to compile the code successfully?

abstract class A

{

int a = 100;

public abstract void showA(); }

public class B extends A

{

\_ \_ \_ \_ \_ \_ \_ \_ \_ // Fill the blank

public static void main(String []args)

{

A objA = new B();

objA.showA();

}

}

|  |  |
| --- | --- |
| **A.** public abstract void showA() { } | **B.** public void showA() { } |
| **C.** void showA() { } | **D.** public B showA() { } |

**Correct Answer : OPTION B, A concrete class extending an abstract class must override abstract method with less restrictive access modifier.**

Q. Which is a valid declaration within an Interface?

|  |  |
| --- | --- |
| **A.** public static short stop = 23; | **B.** protected short stop = 23; |
| **C.** transient short stop = 23; | **D.** final void start(short stop); |

**Correct Answer : OPTION A, public static short stop = 23;. A variable declared inside interface must be public static final variables(constants), it cannot have any other modifier. And Methods inside Interface must not be static, final, native or strictfp.**

Q. Which of the following statement is true about an Interface?

|  |
| --- |
| **A.** Methods inside Interface can be static, final, native or strictfp. |
| **B.** Interface can not extend one or more other interface. |
| **C.** Interface cannot implement a class. |
| **D.** Interface can not be nested inside another interface. |

**Correct Answer : OPTION C, 'Interface cannot implement a class' is the only correct statement from above. Rest all are not true about an Interface.**

Q. Which of following is a valid class using the given code?

public interface A { public void showA(); }

|  |
| --- |
| **A.** public class B extends A { public void showA(){} } |
| **B.** public class B implements A { public abstract void showA(){} } |
| **C.** public class B implements A { void showA(){} } |
| **D.** public class B implements A { public void showA(){} } |

**Correct Answer : OPTION D, Any concrete class implementing an interface must provide implementation (definition) of all the abstract methods of the Interface.**

Q. Given the following declarations, which assignment is legal?

// Class declarations :

interface A {}

class B {}

class C extends B implements A {}

class D implements A {}

// Declaration statements :

B b = new B();

C c = new C();

D d = new D();

|  |  |
| --- | --- |
| **A.** c = d; | **B.** d = c; |
| **C.** A a = d; | **D.** d = (D)c; |

**Correct Answer : OPTION C, A a = d; is the only legal assignment from above code. Here 'a' is the reference of interface A which is pointing to instance of class D. If you define a reference variable whose type is an interface, any object you assign to it must be an instance of a class that implements the interface. So class D instance can be assigned to 'a' because D implements interface A. This is how java support abstraction by hiding the complexity of Class D.**

Q. Collection stores only ?

|  |  |
| --- | --- |
| **A.** object type data | **B.** primitive type data |
| **C.** string type data | **D.** All of the above |

**Correct Answer : OPTION A, object type data. Collections doesn't store primitive data types,it stores only object(refrences). However Autoboxing facilitates the storing of primitive data types in collection by boxing it into its wrapper type.**

Q. Which type of exception is thrown if you try to add an element to an already full Collection ?

|  |  |
| --- | --- |
| **A.** IllegalArgumentException | **B.** IllegalStateException |
| **C.** ClassCastException | **D.** NullPointerException |

**Correct Answer : OPTION B, IllegalStateException is thrown if you try to add an element to an already full Collection.**

Q. Which interface does NavigableSet implements ?

|  |  |
| --- | --- |
| **A.** Set | **B.** List |
| **C.** Collection | **D.** SortedSet |

**Correct Answer : OPTION D, SortedSet**

Q. What does get(int index) method define by List interface do ?

|  |
| --- |
| **A.** returns an object stored at the specified index |
| **B.** stores an object at the specified index |
| **C.** returns a list containing elements between specified index and end in the collection. |
| **D.** none of the above. |

**Correct Answer : OPTION A, get(int index) returns an object stored at the specified index.**

Q. Which of the following interface guarantees that no duplicates elements are stored and all elements are accessed in natural order ?

|  |  |
| --- | --- |
| **A.** java.util.List | **B.** java.util.Set |
| **C.** java.util.Deque | **D.** java.util.Map |

**Correct Answer : OPTION B, java.util.Set**

Q. Which of the following statement is not true about ArrayList class ?

|  |
| --- |
| **A.** ArrayList class extends AbstractList class and implements the List interface |
| **B.** ArrayList supports dynamic array that can grow as needed |
| **C.** ArrayLists are synchronized. |
| **D.** It can coontain Duplicate elements and maintains the insertion order |

**Correct Answer : OPTION C, ArrayLists are synchronized is not a correct statement.**

Q. Which of the following statement is true about TreeSet class ?

|  |
| --- |
| **A.** TreeSet extends AbstractList class and implements the List interface. |
| **B.** Access and retrieval times are quite slow. |
| **C.** TreeSet stores elements sorted ascending order |
| **D.** Uses a LinkedList structure to store elements |

**Correct Answer : OPTION C, TreeSet stores elements sorted ascending order**

Q. Which class does java.util.Properties extends ?

|  |  |
| --- | --- |
| **A.** java.util.HashMap | **B.** java.util.Hashtable |
| **C.** java.util.LinkedHashMap | **D.** java.util.LinkedList |

**Correct Answer : OPTION B, java.util.Hashtable. Properties class extends Hashtable class. It is used to maintain list of value in which both key and value are String.**

Q. Which interface is used to traverse a list in both forward and backward direction ?

|  |  |
| --- | --- |
| **A.** Iterator interface | **B.** ListIterator interface |
| **C.** Enumeration interface | **D.** None of the above |

**Correct Answer : OPTION B, ListIterator Interface is used to traverse a list in both forward and backward direction. It is available to only those collections that implement the List Interface.**

Q. Which interface provides the capability to store objects using a key-value pair ?

|  |  |
| --- | --- |
| **A.** java.util.Map | **B.** java.util.Set |
| **C.** java.util.List | **D.** java.util.Collection |

**Correct Answer : OPTION A, java.util.Map**

Q. Which of the following method is used to return a Set that contains the entries in a map ?

|  |  |
| --- | --- |
| **A.** keyset() | **B.** getSet() |
| **C.** entrySet() | **D.** getAll() |

**Correct Answer : OPTION C, entrySet() method returns a Set that contains the entries in a map**

Q. Which of the following statement is not correct about HashMap class ?

|  |
| --- |
| HashMap class extends AbstractMap and implements Map interface. |
| HashMap class is not synchronized. |
| HashMap uses a hashtable to store the map. |
| HashMap maintain order of its element. |

**Correct Answer : OPTION D, HashMap maintain order of its element statement is not correct.**

Q. Which of the following class provide the facility to specify default value that is used when no value is associated with a certain key ?

|  |  |
| --- | --- |
| java.util.HashMap | java.util.Hashtable |
| java.util.Properties | java.util.LinkedHashMap |

**Correct Answer : OPTION C, java.util.Properties**