Question	Answer 1	Answer 2	Answer 3	Answer 4	Right Answer
String s1="hello"; String s2="hello"; which one will return true?	s1 == s2	s1.equals(s2)	Both I and 2	None of the above	Option2

What will happen in the below code snipet: public class MyClass
{
int i; float f; double d;
boolean bl;
public static void main(String args[])
{
System.out.println("int = "+i);
System.out.println("float = "+f);
System.out.println("double = "+d);
System.out.println("boolean = "+bl);
}
}
	Int=0 float=0.0 double=0.0 boolean=false	Compilation error: cannot make static reference to the non-static field	Int=0 float=0.000 double=0.000 boolean=false	Compilation error: variable may not have been initialized	Option2

What is legal?	Try{}catch()	Try{}catch()finally{}	Try{}finally{}	All of the above	Option4
If u want to create checked exception as user defined exception u need to extend	RuntimeException	Throwable	Exception	Error	Option3
Which of the following is not generally recoverable in the program	Error	Exception	Both a and b	None of the above.	Option1
Java compiler compiles source code into _____ files.	object	shared object	class	0	Option3
There are _____ primitive types in Java type system.	6	8	12	0	Option2
In Java, every user-defined type is _____ type.	always a reference	always a value	either a reference or a value	0	Option1
Java long data-type identifies a _____bit integer value.	16	32	64	0	Option3
In Java the memory for an instance of a reference type is always allocated _____.	on heap	on stack	in data-section	0	Option1
The binary representation of class insurance.Policy is loaded by default from path _____.	./Policy.class	insurance/Policy.class	policy/Insurance.class	0	Option2
A member of a Java class declared _____modifier is visible only to other classes in the same package.	without any	with protected	with public	0	Option1

Question	Answer 1	Answer 2	Answer 3	Answer 4	Right Answer
Native methods are also called as_____	class methods	foreign methods	instance methods	abstract methods	Option2

What will be the output of the following code
public class Myclass
{
public static void main(String args[]){
RuntimeException rte=null;
throw rte;
}
}
	Compile time error, because main method does not declare that it throws RuntimeException in its declaration.	Compile time error, because it cannot throw rte.	The program will compile without an error.	Code will compile without an error, and will throw java.lang. RuntimeException when run.	Option4

What will happen when you attempt to compile or run this code?
class Base
{
public final void amethod ()
{
system.out.println (“amethod”);
}
}
public class Fin extends Base
{
public static void main (String argv [])
{
Base b = new Base() ;
b.amethod () ;
}
}
	Compile time error indicating that a class with any final methods must be declared final itself	Compile time error indicating that you inherit from a class with final methods.	Success in compilation and output of “amethod” at run time	None of the above	Option3

After execution of the following code fragment, what are the values of the variable x, a and b?
int x, a = 6, b = 7;
x = a++ + b++;
	x=15,a=7,b=8	x=15,a=6,b=7	x=13,a=7,b=8	x=13,a=6,b=7	Option3

Question	Answer 1	Answer 2	Answer 3	Answer 4	Right Answer
Which of the following statements is true?	Transient methods may not be overridden.	Transient methods must be overridden.	Transient classes may not be serialized.	Transient variables are not serialized.	Option4
Which of the following statement is true?	An Inner class may be declared private.	An Inner class may be declared static.	Construction of an inner class may require an instance of the outer class.	All of the above	Option4
Which would be most suitable for storing data elements that must not appear in the store more than once, if searching is not a priority?	Collection.	List	Set	Map	Option3
What is the goal of java's serialization facility?	To provide data protection from concurrent access.	To provide data persistence	To provide remote access to data	To improve performance	Option2
Which of the following is the correct syntax for suggesting the JVM performs garbage collection.	System.free ();	System.setGarbageCollection () ;	System.out.get() ;	System.gc();	Option4
What is the correct ordering for the import, class and package declaration when found in a single file?	package, import, class	class, import, package	import, package, class	package, class, import	Option4
String class is	final	abstract	static	transient	Option1
What is the effect of issuing a wait() method on an object	If a notify() method has already been sent to that object then it has no effect	The object issuing the call to wait() will halt until another object sends a notify() or notifyAll() method	An exception will be raised	The object issuing the call to wait() will be automatically synchronized with any other objects using the receiving object.	Option2
static member scope is _____	They are created when the class is loaded at runtime.	They are created when main get called.	They are created when class object get created.	They are created when class get modified	Option1
Wrapper class is part of package_____	java.lang	java.util	java.io	java.awt	Option1

Question	Answer 1	Answer 2	Answer 3	Answer 4	Right Answer

What is wrong with the following code?
abstract class MyClass{
transient int var1;
synchronized int var2;
final void Myclass() {}
static void f() {}
}
	Myclass can not be declared as abstract.	Var1 can not be declared as transient.	Var2 can not be declared as synchronized.	Nothing is wrong in the given line of code.	Option3

Which of the following statement is true?	An object will be garbage collected when it becomes unreachable	An object will be garbage collected if it has null assigned to it	The finalize method will be run before an object is garbage collected	All of the above	Option4
instanceof operator can be used with _____.	Array	Final Class	Classes	All of the above	Option4

public class CDAC{
public static void main(String a[]){
String s1 = "Sun";
System.out.println(s1.substring(5));
} What is output?
	-1	0	StringIndexOutOfBoundsException	ArrayIndexOutOfBoundsException	Option3

Which of the following modifiers can be applied to a constructor?	Private	abstract	volatile	All of the above	Option1

class DAC {
public static void main(String[] s) {
String s1 = "A", s2 = " B ", s3 = "C";
s2.trim(); s3.concat("D");
System.out.print(s1 + s2 + s3);
}
}
What is the result of the program?
	Prints: ABC	Prints: A B C	Prints: ABCD	Prints: ABDC	Option3

a. Entries are organized as key/value pairs b. Duplicate entries replace old entries Which interface of the java.util package offer the specified behaviour?	List	Map	Set	None of the above	Option2

Question	Answer 1	Answer 2	Answer 3	Answer 4	Right Answer

class CDAC {
public static void main (String[] args) {
StringBuffer sb1 = new StringBuffer("ABC");
StringBuffer sb2 = new StringBuffer("ABC");
System.out.print(“Prints:”+(sb1==sb2)+","+sb1.equals(sb2));
}
}
What is the result of attempting to compile and run the program?
	Prints: false, true	Prints: false, false	.
			Prints: true, false
				Prints: true, true	Option1

How many methods are defined in the Cloneable interface?	None	One	Two	None of the above	Option1

class X implements Runnable
{
public static void main(String args[])
{
/* Missing code? */
}
public void run() {}
}
Which of the following line of code is suitable to start
a thread?
	Thread t = new Thread(X);	Thread t = new Thread(X); t.start();	Thread t = new Thread(); x.run();	X run = new X();
				Thread t = new Thread(run);
					Option2

Which method must be defined by a class implementing the java.lang.Runnable interface?	void run()	public void run()	public void start()	void run(int priority)	Option2

What will be the output of the program?
class Equals
{
public static void main(String [] args)
{
int x = 100;
double y = 100.1;
boolean b = (x = y); /* Line 7 */
System.out.println(b);
}
}
	TRUE	FALSE	Compilation fails	An exception is thrown at runtime	Option3

Which of the following is not a Structural Patterns?	Adapter pattern	Bridge Pattern	Command Pattern	Composite Pattern	Option3
Which of the follow are true statements?	An anonymous class can extend only the Object class	An anonymous class can not implement an interface	An anonymous class is implicitly final	An anonymous class can be abstract	Option3