Java Multiple Choice Questions

1.The Java interpreter is used for the execution of the source code.
True
False
Ans: a.
2) On successful compilation a file with the class extension is created.
a) True
b) False
Ans: a.
3) The Java source code can be created in a Notepad editor.
a) True
b) False
Ans: a.
4) The Java Program is enclosed in a class definition.
a) True
b) False
Ans: a.
5) What declarations are required for every Java application?
Ans: A class and the main( ) method declarations.
6) What are the two parts in executing a Java program and their purposes?
Ans: Two parts in executing a Java program are:
Java Compiler and Java Interpreter.
The Java Compiler is used for compilation and the Java Interpreter is used for execution of the application.
7) What are the three OOPs principles and define them?
Ans : Encapsulation, Inheritance and Polymorphism are the three OOPs Principles.
Encapsulation:
Is the Mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse.
Inheritance:
Is the process by which one object acquires the properties of another object.
Polymorphism:
Is a feature that allows one interface to be used for a general class of actions.

8) What is a compilation unit?
Ans : Java source code file.
9) What output is displayed as the result of executing the following statement?
System.out.println("// Looks like a comment.");
// Looks like a comment
The statement results in a compilation error
Looks like a comment
No output is displayed
Ans : a.
10) In order for a source code file, containing the public class Test, to successfully compile, which of the following must be true?
It must have a package statement
It must be named Test.java
It must import java.lang
It must declare a public class named Test
Ans : b
11) What are identifiers and what is naming convention?
Ans : Identifiers are used for class names, method names and variable names. An identifier may be any descriptive sequence of upper case & lower case letters,numbers or underscore or dollar sign and must not begin with numbers.
12) What is the return type of program's main( ) method?

Ans : void
13) What is the argument type of program's main( ) method?
Ans : string array.
14) Which characters are as first characters of an identifier?
Ans : A – Z, a – z, _ ,$
15) What are different comments?
Ans : 1) // -- single line comment
2) /* --
*/ multiple line comment
3) /** --
*/ documentation
16) What is the difference between constructor method and method?
Ans : Constructor will be automatically invoked when an object is created. Whereas method has to be call explicitly.
17) What is the use of bin and lib in JDK?
Ans : Bin contains all tools such as javac, applet viewer, awt tool etc., whereas Lib contains all packages and variables.

Data types,variables and Arrays

1) What is meant by variable?
Ans: Variables are locations in memory that can hold values. Before assigning any value to a variable, it must be declared.
2) What are the kinds of variables in Java? What are their uses?
Ans: Java has three kinds of variables namely, the instance variable, the local variable and the class variable.
Local variables are used inside blocks as counters or in methods as temporary variables and are used to store information needed by a single method.
Instance variables are used to define attributes or the state of a particular object and are used to store information needed by multiple methods in the objects.
Class variables are global to a class and to all the instances of the class and are useful for communicating between different objects of all the same class or keeping track of global states.
3) How are the variables declared?
Ans: Variables can be declared anywhere in the method definition and can be initialized during their declaration.They are commonly declared before usage at the beginning of the definition.
Variables with the same data type can be declared together. Local variables must be given a value before usage.
4) What are variable types?
Ans: Variable types can be any data type that java supports, which includes the eight primitive data types, the name of a class or interface and an array.
5) How do you assign values to variables?
Ans: Values are assigned to variables using the assignment operator =.
6) What is a literal? How many types of literals are there?
Ans: A literal represents a value of a certain type where the type describes how that value behaves. There are different types of literals namely number literals, character literals, boolean literals, string literals,etc.
7) What is an array?
Ans: An array is an object that stores a list of items.
8) How do you declare an array?
Ans: Array variable indicates the type of object that the array holds.
Ex: int arr[];
9) Java supports multidimensional arrays.
a)True
b)False
Ans: a.
10) An array of arrays can be created.

a)True
b)False
Ans: a.
11) What is a string?
Ans: A combination of characters is called as string.
12) Strings are instances of the class String.
a)True
b)False
Ans: a.
13) When a string literal is used in the program, Java automatically creates instances of the string class.
a)True
b)False
Ans: a.
14) Which operator is to create and concatenate string?
Ans: Addition operator(+).
15) Which of the following declare an array of string objects?
String[ ] s;
String [ ]s:
String[s]:
String s[ ]:
Ans : a, b and d
16) What is the value of a[3] as the result of the following array declaration?
1
2
3
4
Ans : d
17) Which of the following are primitive types?
byte
String
integer
Float
Ans : a.
18) What is the range of the char type?
0 to 2^{16}
0 to 2^{15}
0 to 2^{16}-1
0 to 2^{15}-1
Ans. d
19) What are primitive data types?
Ans : byte, short, int, long
float, double
boolean
char
20) What are default values of different primitive types?
Ans : int - 0
short - 0
byte - 0
long - 0 l
float - 0.0 f
double - 0.0 d
boolean - false
char - null
21) Converting of primitive types to objects can be explicitly.
a)True
b)False

Ans: b.
22) How do we change the values of the elements of the array?
Ans : The array subscript expression can be used to change the values of the elements of the array.
23) What is final varaible?
Ans : If a variable is declared as final variable, then you can not change its value. It becomes constant.
24) What is static variable?
Ans : Static variables are shared by all instances of a class.

Operators

1) What are operators and what are the various types of operators available in Java?
Ans: Operators are special symbols used in expressions.
The following are the types of operators:
Arithmetic operators,
Assignment operators,
Increment & Decrement operators,
Logical operators,
Biwise operators,
Comparison/Relational operators and
Conditional operators
2) The ++ operator is used for incrementing and the -- operator is used for decrementing.
a)True
b)False
Ans: a.
3) Comparison/Logical operators are used for testing and magnitude.
a)True
b)False
Ans: a.
4) Character literals are stored as unicode characters.
a)True
b)False
Ans: a.
5) What are the Logical operators?
Ans: OR(|), AND(&), XOR(^) AND NOT(~).
6) What is the % operator?
Ans : % operator is the modulo operator or reminder operator. It returns the reminder of dividing the first operand by second operand.
7) What is the value of 111 % 13?
3
5
7
9
Ans : c.
8) Is &&= a valid operator?
Ans : No.
9) Can a double value be cast to a byte?
Ans : Yes
10) Can a byte object be cast to a double value ?
Ans : No. An object cannot be cast to a primitive value.
11) What are order of precedence and associativity?
Ans : Order of precedence the order in which operators are evaluated in expressions.
Associativity determines whether an expression is evaluated left-right or right-left.
12) Which Java operator is right associativity?
Ans : = operator.

13) What is the difference between prefix and postfix of -- and ++ operators?
Ans : The prefix form returns the increment or decrement operation and returns the value of the increment or decrement operation.
The postfix form returns the current value of all of the expression and then performs the increment or decrement operation on that value.
14) What is the result of expression 5.45 + "3,2"?
The double value 8.6
The string ""8.6"
The long value 8.
The String "5.453.2"
Ans : d
15) What are the values of x and y ?
x = 5; y = ++x;
Ans : x = 6; y = 6
16) What are the values of x and z?
x = 5; z = x++;
Ans : x = 6; z = 5

Control Statements

1) What are the programming constructs?
Ans: a) Sequential
b) Selection -- if and switch statements
c) Iteration -- for loop, while loop and do-while loop
2) class conditional {
public static void main(String args[]) {
int i = 20;
int j = 55;
int z = 0;
z = i < j ? i : j; // ternary operator
System.out.println("The value assigned is " + z);
}
}
What is output of the above program?
Ans: The value assigned is 20
3) The switch statement does not require a break.
a)True
b)False
Ans: b.
4) The conditional operator is otherwise known as the ternary operator.
a)True
b)False
Ans: a.
5) The while loop repeats a set of code while the condition is false.
a)True
b)False
Ans: b.
6) The do-while loop repeats a set of code atleast once before the condition is tested.
a)True
b)False
Ans: a.
7) What are difference between break and continue?

Ans: The break keyword halts the execution of the current loop and forces control out of the loop.
The continue is similar to break, except that instead of halting the execution of the loop, it starts the next iteration.

8) The for loop repeats a set of statements a certain number of times until a condition is matched.
a)True
b)False
Ans: a.
9) Can a for statement loop indefintely?
Ans : Yes.
10) What is the difference between while statement and a do statement/
Ans : A while statement checks at the beginning of a loop to see whether the next loop iteration should occur.
A do statement checks at the end of a loop to see whether the next iteration of a loop should occur. The do statement will always execute the body of a loop at least once.

Introduction to Classes and Methods

1) Which is used to get the value of the instance variables?
Ans: Dot notation.
2) The new operator creates a single instance named class and returns a reference to that object.
a)True
b)False
Ans: a.
3) A class is a template for multiple objects with similar features.
a)True
b)False
Ans: a.
4) What is mean by garbage collection?
Ans: When an object is no longer referred to by any variable, Java automatically reclaims memory used by that object. This is known as garbage collection.
5) What are methods and how are they defined?
Ans: Methods are functions that operate on instances of classes in which they are defined.Objects can communicate with each other using methods and can call methods in other classes.
Method definition has four parts. They are name of the method, type of object or primitive type the method returns, a list of parameters and the body of the method.
A method's signature is a combination of the first three parts mentioned above.
6) What is calling method?
Ans: Calling methods are similar to calling or referring to an instance variable. These methods are accessed using dot notation.
Ex: obj.methodname(param1,param2)
7) Which method is used to determine the class of an object?
Ans: getClass( ) method can be used to find out what class the belongs to. This class is defined in the object class and is available to all objects.
8) All the classes in java.lang package are automatically imported when a program is compiled.
a)True
b)False
Ans: a.
9) How can class be imported to a program?
Ans: To import a class, the import keyword should be used as shown.;
import classname;
10) How can class be imported from a package to a program?
Ans: import java . packagename . classname (or) import java.package name.*;
11) What is a constructor?

Ans: A constructor is a special kind of method that determines how an object is initialized when created.
12) Which keyword is used to create an instance of a class?
Ans: new.
13) Which method is used to garbage collect an object?
Ans: finalize ().
14) Constructors can be overloaded like regular methods.
a)True
b)False
Ans: a.
15) What is casting?
Ans: Casting is bused to convert the value of one type to another.

16) Casting between primitive types allows conversion of one primitive type to another.
a)True
b)False
Ans: a.
17) Casting occurs commonly between numeric types.
a)True
b)False
Ans: a.
18) Boolean values can be cast into any other primitive type.
a)True
b)False
Ans: b.
19) Casting does not affect the original object or value.
a)True
b)False
Ans: a.
20) Which cast must be used to convert a larger value into a smaller one?
Ans: Explicit cast.
21) Which cast must be used to cast an object to another class?
Ans: Specific cast.
22) Which of the following features are common to both Java & C++?
A.The class declaration
b.The access modifiers
c.The encapsulation of data & methods with in objects
d.The use of pointers
Ans: a,b,c.
23) Which of the following statements accurately describe the use of access modifiers within a class definition?
a.They can be applied to both data & methods
b.They must precede a class's data variables or methods
c.They can follow a class's data variables or methods
d.They can appear in any order
e.They must be applied to data variables first and then to methods
Ans: a,b,d.
24) Suppose a given instance variable has been declared private.
Can this instance variable be manipulated by methods out side its class?
a.yes
b.no
Ans: b.
25) Which of the following statements can be used to describe a public method?
a.It is accessible to all other classes in the hierarchy

b.It is accessablde only to subclasses of its parent class
c.It represents the public interface of its class
d.The only way to gain access to this method is by calling one of the public class methods
Ans: a,c.
26) Which of the following types of class members can be part of the internal part of a class?
a.Public instance variables
b.Private instance variables
c.Public methods
d.Private methods
Ans: b,d.
27) You would use the ____ operator to create a single instance of a named class.
a.new
b.dot
Ans: a.
28) Which of the following statements correctly describes the relation between an object and the instance variable it stores?
a.Each new object has its own distinctive set of instance variables
b.Each object has a copy of the instance variables of its class
c.the instance variable of each object are seperate from the variables of other objects
d.The instance variables of each object are stored together with the variables of other objects
Ans: a,b,c.
29) If no input parameters are specified in a method declaration then the declaration will include __.
a.an empty set of parantheses
b.the term void
Ans: a.
30) What are the functions of the dot(.) operator?
a.It enables you to access instance variables of any objects within a class
b.It enables you to store values in instance variables of an object
c.It is used to call object methods
d.It is to create a new object
Ans: a,b,c.
31) Which of the following can be referenced by this variable?
a.The instance variables of a class only
b.The methods of a class only
c.The instance variables and methods of a class
Ans: c.
32) The this reference is used in conjunction with ___methods.
a.static
b.non-static
Ans: b.
33) Which of the following operators are used in conjunction with the this and super references?
a.The new operator
b.The instanceof operator
c.The dot operator
Ans: c.
34) A constructor is automatically called when an object is instantiated
a. true
b. false
Ans: a.
35) When may a constructor be called without specifying arguments?
a. When the default constructor is not called
b. When the name of the constructor differs from that of the class
c. When there are no constructors for the class
Ans: c.
36) Each class in java can have a finalizer method

a. true
b.false
Ans: a.
37) When an object is referenced, does this mean that it has been identified by the finalizer method for garbage collection?
a.yes
b.no
Ans: b.
38) Because finalize () belongs to the java.lang.Object class, it is present in all ___.
a.objects
b.classes
c.methods
Ans: b.
39) Identify the true statements about finalization.
a.A class may have only one finalize method
b.Finalizers are mostly used with simple classes
c.Finalizer overloading is not allowed
Ans: a,c.
40) When you write finalize() method for your class, you are overriding a finalizer inherited from a super class.
a.true
b.false
Ans: a.
41) Java memory management mechanism garbage collects objects which are no longer referenced
a true
b.false
Ans: a.
42) are objects referenced by a variable candidates for garbage collection when the variable goes out of scope?
a yes
b. no
Ans: a.
43) Java's garbage collector runs as a ___ priority thread waiting for __priority threads to relinquish the processor.
a.high
b.low
Ans: a,b.
44) The garbage collector will run immediately when the system is out of memory
a.true
b.false
Ans: a.
45) You can explicitly drop a object reference by setting the value of a variable whose data type is a reference type to ___
Ans: null
46) When might your program wish to run the garbage collecter?
a. before it enters a compute-intense section of code
b. before it enters a memory-intense section of code
c. before objects are finalized
d. when it knows there will be some idle time
Ans: a,b,d
47) For externalizable objects the class is solely responsible for the external format of its contents
a.true
b.false
Ans: a
48) When an object is stored, are all of the objects that are reachable from that object stored as well?
a.true

b.false
Ans: a
49) The default__ of objects protects private and trancient data, and supports the __ of the classes
a.evolution
b.encoding
Ans: b,a.
50) Which are keywords in Java?
a) NULL
b) sizeof
c) friend
d) extends
e) synchronized
Ans : d and e
51) When must the main class and the file name coincide?
Ans :When class is declared public.
52) What are different modifiers?
Ans : public, private, protected, default, static, trancient, volatile, final, abstract.
53) What are access modifiers?
Ans : public, private, protected, default.
54) What is meant by "Passing by value" and " Passing by reference"?
Ans : objects – pass by referrence
Methods - pass by value
55) Is a class a subclass of itself?
Ans : A class is a subclass itself.

56) What modifiers may be used with top-level class?
Ans : public, abstract, final.
57) What is an example of polymorphism?
Inner class
Anonymous classes
Method overloading
Method overriding
Ans : c

Packages and interface

1) What are packages ? what is use of packages ?
Ans :The package statement defines a name space in which classes are stored.If you omit the package, the classes are put into the default package.
Signature... package pkg;
Use: * It specifies to which package the classes defined in a file belongs to. * Package is both naming and a visibility control mechanism.
2) What is difference between importing "java.applet.Applet" and "java.applet.*;" ?
Ans :"java.applet.Applet" will import only the class Applet from the package java.applet
Where as "java.applet.*" will import all the classes from java.applet package.
3) What do you understand by package access specifier?
Ans : public: Anything declared as public can be accessed from anywhere
private: Anything declared in the private can't be seen outside of its class.
default: It is visible to subclasses as well as to other classes in the same package.
4) What is interface? What is use of interface?
Ans : It is similar to class which may contain method's signature only but not bodies.
Methods declared in interface are abstract methods. We can implement many interfaces on a class which support the multiple inheritance.
5) Is it is necessary to implement all methods in an interface?
Ans : Yes. All the methods have to be implemented.
6) Which is the default access modifier for an interface method?

Ans : public.
7) Can we define a variable in an interface ?and what type it should be ?
Ans : Yes we can define a variable in an interface. They are implicitly final and static.
8) What is difference between interface and an abstract class?
Ans : All the methods declared inside an Interface are abstract. Where as abstract class must have at least one abstract method and others may be concrete or abstract.
In Interface we need not use the keyword abstract for the methods.
9) By default, all program import the java.lang package.
True/False
Ans : True
10) Java compiler stores the .class files in the path specified in CLASSPATH environmental variable.
True/False
Ans : False

11) User-defined package can also be imported just like the standard packages.
True/False
Ans : True
12) When a program does not want to handle exception, the _____class is used.
Ans : Throws
13) The main subclass of the Exception class is _____ class.
Ans : RuntimeException
14) Only subclasses of _____class may be caught or thrown.
Ans : Throwable
15) Any user-defined exception class is a subclass of the _____ class.
Ans : Exception
16) The catch clause of the user-defined exception class should _____ its Base class catch clause.
Ans : Exception
17) A _____ is used to separate the hierarchy of the class while declaring an Import statement.
Ans : Package

18) All standard classes of Java are included within a package called _____.
Ans : java.lang
19) All the classes in a package can be simultaneously imported using ____.
Ans : *
20) Can you define a variable inside an Interface. If no, why? If yes, how?
Ans.: YES. final and static
21) How many concrete classes can you have inside an interface?
Ans.: None
22) Can you extend an interface?
Ans.: Yes
23) Is it necessary to implement all the methods of an interface while implementing the interface?
Ans.: No
24) If you do not implement all the methods of an interface while implementing , what specifier should you use for the class ?
Ans.: abstract
25) How do you achieve multiple inheritance in Java?
Ans: Using interfaces.
26) How to declare an interface example?
Ans : access class classname implements interface.
27) Can you achieve multiple interface through interface?
a)True
b) false
Ans : a.

28) Can variables be declared in an interface ? If so, what are the modifiers?
Ans : Yes. final and static are the modifiers can be declared in an interface.
29) What are the possible access modifiers when implementing interface methods?
Ans : public.
30) Can anonymous classes be implemented an interface?
Ans : Yes.
31) Interfaces can't be extended.
a)True
b)False
Ans : b.
32) Name interfaces without a method?
Ans : Serializable, Cloneble & Remote.
33) Is it possible to use few methods of an interface in a class ? If so, how?
Ans : Yes. Declare the class as abstract.

Exception Handling

1) What is the difference between 'throw' and 'throws' ?And it's application?
Ans : Exceptions that are thrown by java runtime systems can be handled by Try and catch blocks. With throw exception we can handle the exceptions thrown by the program itself. If a method is capable of causing an exception that it does not
handle, it must specify this behavior so the callers of the method can guard
against that exception.
2) What is the difference between 'Exception' and 'error' in java?
Ans : Exception and Error are the subclasses of the Throwable class. Exception class is used for exceptional conditions that user program should catch. With exception class we can subclass to create our own custom exception.
Error defines exceptions that are not excepted to be caught by you program. Example is Stack Overflow.
3) What is 'Resource leak'?
Ans : Freeing up other resources that might have been allocated at the beginning of a method.
4)What is the 'finally' block?
Ans : Finally block will execute whether or not an exception is thrown. If an exception is thrown, the finally block will execute even if no catch statement match the exception. Any time a method is about to return to the caller from inside try/catch block, via an uncaught exception or an explicit return statement, the finally clause is also execute.
5) Can we have catch block with out try block? If so when?
Ans : No. Try/Catch or Try/finally form a unit.
6) What is the difference between the following statements?
Catch (Exception e),
Catch (Error err),
Catch (Throwable t)
Ans :

7) What will happen to the Exception object after exception handling?
Ans : It will go for Garbage Collector. And frees the memory.
8) How many Exceptions we can define in 'throws' clause?
Ans : We can define multiple exceptions in throws clause.
Signature is..
type method-name (parameter-list) throws exception-list

9) The finally block is executed when an exception is thrown, even if no catch matches it.
True/False

Ans : True
10) The subclass exception should precede the base class exception when used within the catch clause.
True/False
Ans : True
11) Exceptions can be caught or rethrown to a calling method.
True/False
Ans : True
12) The statements following the throw keyword in a program are not executed.
True/False
Ans : True
13) The toString ( ) method in the user-defined exception class is overridden.
True/False
Ans : True

MULTI THREADING

1) What are the two types of multitasking?
Ans : 1.process-based
2.Thread-based
2) What are the two ways to create the thread?
Ans : 1.by implementing Runnable
2.by extending Thread
3) What is the signature of the constructor of a thread class?
Ans : Thread(Runnable threadob,String threadName)
4) What are all the methods available in the Runnable Interface?
Ans : run()
5) What is the data type for the method isAlive() and this method is
available in which class?
Ans : boolean, Thread
6) What are all the methods available in the Thread class?
Ans : 1.isAlive()
2.join()
3.resume()
4.suspend()
5.stop()
6.start()
7.sleep()
8.destroy()
7) What are all the methods used for Inter Thread communication and what is the class in which these methods are defined?
Ans :1. wait(),notify() & notifyall()
2. Object class
8) What is the mechanisam defind by java for the Resources to be used by only one Thread at a time?
Ans : Synchronisation
9) What is the procedure to own the moniter by many threads?
Ans : not possible

10) What is the unit for 1000 in the below statement?
ob.sleep(1000)
Ans : long milliseconds
11) What is the data type for the parameter of the sleep() method?
Ans : long
12) What are all the values for the following level?
max-priority
min-priority
normal-priority
Ans : 10,1,5
13) What is the method available for setting the priority?
Ans : setPriority()
14) What is the default thread at the time of starting the program?
Ans : main thread
15) The word synchronized can be used with only a method.
True/ False
Ans : False
16) Which priority Thread can prompt the lower primary Thread?
Ans : Higher Priority
17) How many threads at a time can access a monitor?
Ans : one
18) What are all the four states associated in the thread?
Ans : 1. new 2. runnable 3. blocked 4. dead
19) The suspend()method is used to teriminate a thread?
True /False
Ans : False
20) The run() method should necessary exists in clases created as subclass of thread?
True /False
Ans : True
21) When two threads are waiting on each other and can't proceed the programe is said to be in a deadlock?
True/False
Ans : True
22) Which method waits for the thread to die ?
Ans : join() method

23) Which of the following is true?
1) wait(),notify(),notifyall() are defined as final & can be called only from with in a synchronized method
2) Among wait(),notify(),notifyall() the wait() method only throws IOException
3) wait(),notify(),notifyall() & sleep() are methods of object class
1
2
3
1 & 2
1,2 & 3
Ans : D
24) Garbage collector thread belongs to which priority?
Ans : low-priority
25) What is meant by timeslicing or time sharing?
Ans : Timeslicing is the method of allocating CPU time to individual threads in a priority schedule.
26) What is meant by daemon thread? In java runtime, what is it's role?
Ans : Daemon thread is a low priority thread which runs intermittently in the background doing the garbage collection operation for the java runtime system.

Inheritance

1) What is the difference between superclass & subclass?
Ans : A super class is a class that is inherited whereas subclass is a class that does the inheriting.
2) Which keyword is used to inherit a class?
Ans : extends
3) Subclasses methods can access superclass members/ attributes at all times?
True/False
Ans : False

4) When can subclasses not access superclass members?
Ans : When superclass is declared as private.
5) Which class does begin Java class hierarchy?
Ans : Object class
6) Object class is a superclass of all other classes?
True/False
Ans : True
7) Java supports multiple inheritance?
True/False
Ans : False
8) What is inheritance?
Ans : Deriving an object from an existing class. In the other words, Inheritance is the process of inheriting all the features from a class
9) What are the advantages of inheritance?
Ans : Reusability of code and accessibility of variables and methods of the superclass by subclasses.
10) Which method is used to call the constructors of the superclass from the subclass?
Ans : super(argument)
11) Which is used to execute any method of the superclass from the subclass?
Ans : super.method-name(arguments)
12) Which methods are used to destroy the objects created by the constructor methods?
Ans : finalize()
13) What are abstract classes?
Ans : Abstract classes are those for which instances can't be created.
14) What must a class do to implement an interface?
Ans: It must provide all of the methods in the interface and identify the interface in its implements clause.
15) Which methods in the Object class are declared as final?
Ans : getClass(), notify(), notifyAll(), and wait()
16) Final methods can be overridden.
True/False
Ans : False
17) Declaration of methods as final results in faster execution of the program?
True/False
Ans: True
18) Final variables should be declared in the beginning?
True/False
Ans : True
19) Can we declare variable inside a method as final variables? Why?
Ans : Cannot because, local variable cannot be declared as final variables.
20) Can an abstract class may be final?
Ans : An abstract class may not be declared as final.
21) Does a class inherit the constructors of it's super class?
Ans: A class does not inherit constructors from any of it's super classes.
22) What restrictions are placed on method overloading?
Ans: Two methods may not have the same name and argument list but different return types.

23) What restrictions are placed on method overriding?
Ans : Overridden methods must have the same name , argument list , and return type. The overriding method may not limit the access of the method it overridees.The overriding method may not throw any exceptions that may not be thrown by the overridden method.
24) What modifiers may be used with an inner class that is a member of an outer class?
Ans : a (non-local) inner class may be declared as public, protected, private, static, final or abstract.
25) How this() is used with constructors?
Ans: this() is used to invoke a constructor of the same class
26) How super() used with constructors?
Ans : super() is used to invoke a super class constructor
27) Which of the following statements correctly describes an interface?
a)It's a concrete class
b)It's a superclass
c)It's a type of abstract class
Ans: c
28) An interface contains __ methods
a)Non-abstract
b)Implemented
c)unimplemented
Ans:c

STRING HANDLING

Which package does define String and StringBuffer classes?
Ans : java.lang package.
Which method can be used to obtain the length of the String?
Ans : length( ) method.
How do you concatenate Strings?
Ans : By using " + " operator.
Which method can be used to compare two strings for equality?
Ans : equals( ) method.
Which method can be used to perform a comparison between strings that ignores case differences?
Ans : equalsIgnoreCase( ) method.
What is the use of valueOf( ) method?
Ans : valueOf( ) method converts data from its internal format into a human-readable form.
What are the uses of toLowerCase( ) and toUpperCase( ) methods?
Ans : The method toLowerCase( ) converts all the characters in a string from uppercase to lowercase.
The method toUpperCase( ) converts all the characters in a string from lowercase to uppercase.
Which method can be used to find out the total allocated capacity of a StrinBuffer?
Ans : capacity( ) method.
Which method can be used to set the length of the buffer within a StringBuffer object?
Ans : setLength( ).
What is the difference between String and StringBuffer?
Ans : String objects are constants, whereas StringBuffer objects are not.
String class supports constant strings, whereas StringBuffer class supports growable, modifiable strings.
What are wrapper classes?

Ans : Wrapper classes are classes that allow primitive types to be accessed as objects.
Which of the following is not a wrapper class?
String
Integer
Boolean
Character
Ans : a.
What is the output of the following program?

```
public class Question {
public static void main(String args[]) {
String s1 = "abc";
String s2 = "def";
String s3 = s1.concat(s2.toUpperCase( ) );
System.out.println(s1+s2+s3);
}
}
```

abcdefabcdef
abcabcDEFDEF
abcdefabcDEF
None of the above
ANS : c.
Which of the following methods are methods of the String class?
delete( )
append( )
reverse( )
replace( )
Ans : d.
Which of the following methods cause the String object referenced by s to be changed?
s.concat( )
s.toUpperCase( )
s.replace( )
s.valueOf( )
Ans : a and b.
String is a wrapper class?
True
False
Ans : b.
17) If you run the code below, what gets printed out?

```
String s=new String("Bicycle");

int iBegin=1;

char iEnd=3;

System.out.println(s.substring(iBegin,iEnd));
```

Bic
ic
c) icy
d) error: no method matching substring(int,char)
Ans : b.
18) Given the following declarations

```
String s1=new String("Hello")

String s2=new String("there");

String s3=new String();
```

Which of the following are legal operations?
s3=s1 + s2;
s3=s1 - s2;
c) s3=s1 & s2
d) s3=s1 && s2
Ans : a.
19) Which of the following statements are true?
The String class is implemented as a char array, elements are addressed using the stringname[] convention
b) Strings are a primitive type in Java that overloads the + operator for concatenation
c) Strings are a primitive type in Java and the StringBuffer is used as the matching wrapper type
d) The size of a string can be retrieved using the length property.
Ans : b.

EXPLORING JAVA.LANG
java.lang package is automatically imported into all programs.
True
False
Ans : a
What are the interfaces defined by java.lang?
Ans : Cloneable, Comparable and Runnable.
What are the constants defined by both Flaot and Double classes?
Ans : MAX_VALUE,
MIN_VALUE,
NaN,
POSITIVE_INFINITY,
NEGATIVE_INFINITY and
TYPE.
What are the constants defined by Byte, Short, Integer and Long?
Ans : MAX_VALUE,
MIN_VALUE and
TYPE.
What are the constants defined by both Float and Double classes?
Ans : MAX_RADIX,
MIN_RADIX,
MAX_VALUE,
MIN_VALUE and
TYPE.
What is the purpose of the Runtime class?
Ans : The purpose of the Runtime class is to provide access to the Java runtime system.
What is the purpose of the System class?
Ans : The purpose of the System class is to provide access to system resources.
Which class is extended by all other classes?
Ans : Object class is extended by all other classes.
Which class can be used to obtain design information about an object?
Ans : The Class class can be used to obtain information about an object's design.
Which method is used to calculate the absolute value of a number?
Ans : abs( ) method.
What are E and PI?
Ans : E is the base of the natural logarithm and PI is the mathematical value pi.
Which of the following classes is used to perform basic console I/O?
System
SecurityManager
Math

Runtime
Ans : a.
Which of the following are true?
The Class class is the superclass of the Object class.
The Object class is final.
The Class class can be used to load other classes.
The ClassLoader class can be used to load other classes.
Ans : c and d.
Which of the following methods are methods of the Math class?
absolute( )
log( )
cosine( )
sine( )
Ans : b.
Which of the following are true about the Error and Exception classes?
Both classes extend Throwable.
The Error class is final and the Exception class is not.
The Exception class is final and the Error is not.
Both classes implement Throwable.
Ans : a.
Which of the following are true?
The Void class extends the Class class.
The Float class extends the Double class.
The System class extends the Runtime class.
The Integer class extends the Number class.
Ans : d.

17) Which of the following will output -4.0
System.out.println(Math.floor(-4.7));
System.out.println(Math.round(-4.7));
System.out.println(Math.ceil(-4.7));
d) System.out.println(Math.Min(-4.7));
Ans : c.
18) Which of the following are valid statements
a) public class MyCalc extends Math
b) Math.max(s);
c) Math.round(9.99,1);
d) Math.mod(4,10);
e) None of the above.
Ans : e.
19) What will happen if you attempt to compile and run the following code?

```
Integer ten=new Integer(10);

Long nine=new Long (9);

System.out.println(ten + nine);

int i=1;

System.out.println(i + ten);
```

19 followed by 20
19 followed by 11
Error: Can't convert java lang Integer
d) 10 followed by 1

Ans : c.

INPUT / OUTPUT : EXPLORING JAVA.IO

What is meant by Stream and what are the types of Streams and classes of the Streams?
Ans : A Stream is an abstraction that either produces or consumes information.
There are two types of Streams. They are:
Byte Streams : Byte Streams provide a convenient means for handling input and output of bytes.
Character Streams : Character Streams provide a convenient means for handling input and output of characters.
Byte Stream classes : Byte Streams are defined by using two abstract classes. They are:InputStream and OutputStream.
Character Stream classes : Character Streams are defined by using two abstract classes. They are : Reader and Writer.
Which of the following statements are true?
UTF characters are all 8-bits.
UTF characters are all 16-bits.
UTF characters are all 24-bits.
Unicode characters are all 16-bits.
Bytecode characters are all 16-bits.
Ans : d.
Which of the following statements are true?
When you construct an instance of File, if you do not use the filenaming semantics of the local machine, the constructor will throw an IOException.
When you construct an instance of File, if the corresponding file does not exist on the local file system, one will be created.
When an instance of File is garbage collected, the corresponding file on the local file system is deleted.
None of the above.
Ans : a,b and c.
The File class contains a method that changes the current working directory.
True
False
Ans : b.
It is possible to use the File class to list the contents of the current working directory.
True
False
Ans : a.
Readers have methods that can read and return floats and doubles.
True
False
Ans : b.
You execute the code below in an empty directory. What is the result?
File f1 = new File("dirname");
File f2 = new File(f1, "filename");
A new directory called dirname is created in the current working directory.
A new directory called dirname is created in the current working directory. A new file called filename is created in directory dirname.
A new directory called dirname and a new file called filename are created, both in the current working directory.
A new file called filename is created in the current working directory.
No directory is created, and no file is created.
Ans : e.
What is the difference between the Reader/Writer class hierarchy and the InputStream/OutputStream class hierarchy?
Ans : The Reader/Writer class hierarchy is character-oriented and the InputStream/OutputStream class hierarchy is byte-oriented.
What is an I/O filter?

Ans : An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another.

What is the purpose of the File class?

Ans : The File class is used to create objects that provide access to the files and directories of a local file system.

What interface must an object implement before it can be written to a stream as an object?

Ans : An object must implement the Serializable or Externalizable interface before it can be written to a stream as an object.

What is the difference between the File and RandomAccessFile classes?

Ans : The File class encapsulates the files and directories of the local file system. The RandomAccessFile class provides the methods needed to directly access data contained in any part of a file.

What class allows you to read objects directly from a stream?

Ans : The ObjectInputStream class supports the reading of objects from input streams.

What value does read( ) return when it has reached the end of a file?

Ans : The read( ) method returns – 1 when it has reached the end of a file.

What value does readLine( ) return when it has reached the end of a file?

Ans : The readLine( ) method returns null when it has reached the end of a file.

How many bits are used to represent Unicode, ASCII, UTF-16 and UTF-8 characters?

Ans : Unicode requires 16-bits and ASCII requires 8-bits. Although the ASCII character set uses only 1-bits, it is usually represented as 8-bits. UTF-8 represents characters using 8, 16 and 18-bit patterns. UTF-16 uses 16-bit and larger bit patterns.

Which of the following are true?

The InputStream and OutputStream classes are byte-oriented.
The ObjectInputStream and ObjectOutputStream do not support serialized object input and output.
The Reader and Writer classes are character-oriented.
The Reader and Writer classes are the preferred solution to serialized object output.

Ans : a and c.

Which of the following are true about I/O filters?

Filters are supported on input, but not on output.
Filters are supported by the InputStream/OutputStream class hierarchy, but not by the Reader/Writer class hierarchy.
Filters read from one stream and write to another.
A filter may alter data that is read from one stream and written to another.

Ans : c and d.

Which of the following are true?

Any Unicode character is represented using 16-bits.
7-bits are needed to represent any ASCII character.
UTF-8 characters are represented using only 8-bits.
UTF-16 characters are represented using only 16-bits.

Ans : a and b.

Which of the following are true?

The Serializable interface is used to identify objects that may be written to an output stream.
The Externalizable interface is implemented by classes that control the way in which their objects are serialized.
The Serializable interface extends the Externalizable interface.
The Externalizable interface extends the Serializable interface.

Ans : a, b and d.

Which of the following are true about the File class?

A File object can be used to change the current working directory.
A File object can be used to access the files in the current directory.
When a File object is created, a corresponding directory or file is created in the local file system.
File objects are used to access files and directories on the local file system.
File objects can be garbage collected.
When a File object is garbage collected, the corresponding file or directory is deleted.

Ans : b, d and e.

How do you create a Reader object from an InputStream object?
Use the static createReader( ) method of InputStream class.
Use the static createReader( ) method of Reader class.
Create an InputStreamReader object, passing the InputStream object as an argument to the InputStreamReader constructor.
Create an OutputStreamReader object, passing the InputStream object as an argument to the OutputStreamReader constructor.
Ans : c.
Which of the following are true?
Writer classes can be used to write characters to output streams using different character encodings.
Writer classes can be used to write Unicode characters to output streams.
Writer classes have methods that support the writing of the values of any Java primitive type to output streams.
Writer classes have methods that support the writing of objects to output streams.
Ans : a and b.
The isFile( ) method returns a boolean value depending on whether the file object is a file or a directory.
True.
False.
Ans : a.
Reading or writing can be done even after closing the input/output source.
True.
False.
Ans : b.

The _____ method helps in clearing the buffer.
Ans : flush( ).
The System.err method is used to print error message.
True.
False.
Ans : a.
What is meant by StreamTokenizer?
Ans : StreamTokenizer breaks up InputStream into tokens that are delimited by sets of characters.
It has the constructor : StreamTokenizer(Reader inStream).
Here inStream must be some form of Reader.
What is Serialization and deserialization?
Ans : Serialization is the process of writing the state of an object to a byte stream.
Deserialization is the process of restoring these objects.
30) Which of the following can you perform using the File class?
a) Change the current directory
b) Return the name of the parent directory
c) Delete a file
d) Find if a file contains text or binary information
Ans : b and c.
31)How can you change the current working directory using an instance of the File class called FileName?
FileName.chdir("DirName").
FileName.cd("DirName").
FileName.cwd("DirName").
The File class does not support directly changing the current directory.
Ans : d.

EVENT HANDLING

The event delegation model, introduced in release 1.1 of the JDK, is fully compatible with the event model.

True

False

Ans : b.

A component subclass that has executed enableEvents( ) to enable processing of a certain kind of event cannot also use an adapter as a listener for the same kind of event.
True
False
Ans : b.
What is the highest-level event class of the event-delegation model?
Ans : The java.util.eventObject class is the highest-level class in the event-delegation hierarchy.
What interface is extended by AWT event listeners?
Ans : All AWT event listeners extend the java.util.EventListener interface.
What class is the top of the AWT event hierarchy?
Ans : The java.awt.AWTEvent class is the highest-level class in the AWT event class hierarchy.
What event results from the clicking of a button?
Ans : The ActionEvent event is generated as the result of the clicking of a button.
What is the relationship between an event-listener interface and an event-adapter class?
Ans : An event-listener interface defines the methods that must be implemented by an event handler for a particular kind of event.
An event adapter provides a default implementation of an event-listener interface.
In which package are most of the AWT events that support the event-delegation model defined?
Ans : Most of the AWT–related events of the event-delegation model are defined in the java.awt.event package. The AWTEvent class is defined in the java.awt package.
What is the advantage of the event-delegation model over the earlier event-inheritance model?
Ans : The event-delegation has two advantages over the event-inheritance model. They are :
It enables event handling by objects other than the ones that generate the events. This allows a clean separation between a component's design and its use.
It performs much better in applications where many events are generated. This performance improvement is due to the fact that the event-delegation model does not have to repeatedly process unhandled events, as is the case of the event-inheritance model.
What is the purpose of the enableEvents( ) method?
Ans :The enableEvents( ) method is used to enable an event for a particular object.
Which of the following are true?
The event-inheritance model has replaced the event-delegation model.
The event-inheritance model is more efficient than the event-delegation model.
The event-delegation model uses event listeners to define the methods of event-handling classes.
The event-delegation model uses the handleEvent( ) method to support event handling.
Ans : c.
Which of the following is the highest class in the event-delegation model?
java.util.EventListener
java.util.EventObject
java.awt.AWTEvent
java.awt.event.AWTEvent
Ans : b.
When two or more objects are added as listeners for the same event, which listener is first invoked to handle the event?
The first object that was added as listener.
The last object that was added as listener.
There is no way to determine which listener will be invoked first.
It is impossible to have more than one listener for a given event.
Ans : c.
Which of the following components generate action events?
Buttons
Labels
Check boxes
Windows
Ans : a.
Which of the following are true?

A TextField object may generate an ActionEvent.
A TextArea object may generate an ActionEvent.
A Button object may generate an ActionEvent.
A MenuItem object may generate an ActionEvent.
Ans : a,c and d.
Which of the following are true?
The MouseListener interface defines methods for handling mouse clicks.
The MouseMotionListener interface defines methods for handling mouse clicks.
The MouseClickListener interface defines methods for handling mouse clicks.
The ActionListener interface defines methods for handling the clicking of a button.
Ans : a and d.
Suppose that you want to have an object eh handle the TextEvent of a TextArea object t. How should you add eh as the event handler for t?
t.addTextListener(eh);
eh.addTextListener(t);
addTextListener(eh.t);
addTextListener(t,eh);
Ans : a.
What is the preferred way to handle an object's events in Java 2?
Override the object's handleEvent( ) method.
Add one or more event listeners to handle the events.
Have the object override its processEvent( ) methods.
Have the object override its dispatchEvent( ) methods.
Ans : b.
Which of the following are true?
A component may handle its own events by adding itself as an event listener.
A component may handle its own events by overriding its event-dispatching method.
A component may not handle oits own events.
A component may handle its own events only if it implements the handleEvent( ) method.
Ans : a and b.

APPLETS

What is an Applet? Should applets have constructors?
Ans : Applet is a dynamic and interactive program that runs inside a Web page
displayed by a Java capable browser. We don't have the concept of Constructors in Applets.
How do we read number information from my applet's parameters, given that Applet's getParameter() method returns a string?
Ans : Use the parseInt() method in the Integer Class, the Float(String) constructor in the

Class Float, or the Double(String) constructor in the class Double.
How can I arrange for different applets on a web page to communicate with each other?
Ans : Name your applets inside the Applet tag and invoke AppletContext's getApplet() method in your applet code to obtain references to the other applets on the page.
How do I select a URL from my Applet and send the browser to that page?
Ans : Ask the applet for its applet context and invoke showDocument() on that context object.
Eg. URL targetURL;
String URLString
AppletContext context = getAppletContext();
try{
targetUR L = new URL(URLString);
} catch (Malformed URLException e){
// Code for recover from the exception
}
context. showDocument (targetURL);
Can applets on different pages communicate with each other?
Ans : No. Not Directly. The applets will exchange the information at one meeting place either on the local file system or at remote system.
How do Applets differ from Applications?
Ans : Appln: Stand Alone
Applet: Needs no explicit installation on local m/c.
Appln: Execution starts with main() method.
Applet: Execution starts with init() method.
Appln: May or may not be a GUI
Applet: Must run within a GUI (Using AWT)
How do I determine the width and height of my application?
Ans : Use the getSize() method, which the Applet class inherits from the Component class in the Java.awt package. The getSize() method returns the size of the applet as a Dimension object, from which you extract separate width, height fields.
Eg. Dimension dim = getSize ();
int appletwidth = dim.width ();
8) What is AppletStub Interface?
Ans : The applet stub interface provides the means by which an applet and the browser communicate. Your code will not typically implement this interface.
It is essential to have both the .java file and the .html file of an applet in the same directory.
True.
False.
Ans : b.
The <PARAM> tag contains two attributes namely _____ and _____.
Ans : Name , value.

Passing values to parameters is done in the _____ file of an applet.
Ans : .html.
12) What tags are mandatory when creating HTML to display an applet
name, height, width
code, name
codebase, height, width
d) code, height, width
Ans : d.
Applet's getParameter( ) method can be used to get parameter values.
True.
False.
Ans : a.
What are the Applet's Life Cycle methods? Explain them?
Ans : init( ) method - Can be called when an applet is first loaded.

start( ) method - Can be called each time an applet is started.
paint( ) method - Can be called when the applet is minimized or refreshed.
stop( ) method - Can be called when the browser moves off the applet's page.
destroy( ) method - Can be called when the browser is finished with the applet.
What are the Applet's information methods?
Ans : getAppletInfo( ) method : Returns a string describing the applet, its author ,copy right information, etc.
getParameterInfo( ) method : Returns an array of string describing the applet's parameters.
All Applets are subclasses of Applet.
True.
False.
Ans : a.
All Applets must import java.applet and java.awt.
True.
False.
Ans : a.
What are the steps involved in Applet development?
Ans : a) Edit a Java source file,
b) Compile your program and
c) Execute the appletviewer, specifying the name of your applet's source file.
Applets are executed by the console based Java run-time interpreter.
True.
False.
Ans : b.
Which classes and interfaces does Applet class consist?
Ans : Applet class consists of a single class, the Applet class and three interfaces: AppletContext, AppletStub and AudioClip.
What is the sequence for calling the methods by AWT for applets?
Ans : When an applet begins, the AWT calls the following methods, in this sequence.
init( )
start( )
paint( )
When an applet is terminated, the following sequence of method cals takes place :
stop( )
destroy( )
Which method is used to output a string to an applet?
Ans : drawString ( ) method.
Every color is created from an RGB value.
True.
False
Ans : a.

AWT : WINDOWS, GRAPHICS AND FONTS
How would you set the color of a graphics context called g to cyan?
g.setColor(Color.cyan);
g.setCurrentColor(cyan);
g.setColor("Color.cyan");
g.setColor("cyan');
g.setColor(new Color(cyan));
Ans : a.
The code below draws a line. What color is the line?
g.setColor(Color.red.green.yellow.red.cyan);
g.drawLine(0, 0, 100,100);
Red
Green

Yellow
Cyan
Black
Ans : d.
What does the following code draw?
g.setColor(Color.black);
g.drawLine(10, 10, 10, 50);
g.setColor(Color.RED);
g.drawRect(100, 100, 150, 150);
A red vertical line that is 40 pixels long and a red square with sides of 150 pixels
A black vertical line that is 40 pixels long and a red square with sides of 150 pixels
A black vertical line that is 50 pixels long and a red square with sides of 150 pixels
A red vertical line that is 50 pixels long and a red square with sides of 150 pixels
A black vertical line that is 40 pixels long and a red square with sides of 100 pixel
Ans : b.
Which of the statements below are true?
A polyline is always filled.
b) A polyline can not be filled.
c) A polygon is always filled.
d) A polygon is always closed
e) A polygon may be filled or not filled
Ans : b, d and e.
What code would you use to construct a 24-point bold serif font?
new Font(Font.SERIF, 24,Font.BOLD);
new Font("SERIF", 24, BOLD");
new Font("BOLD ", 24,Font.SERIF);
new Font("SERIF", Font.BOLD,24);
new Font(Font.SERIF, "BOLD", 24);
Ans : d.
What does the following paint( ) method draw?
Public void paint(Graphics g) {
g.drawString("question #6",10,0);
}
The string "question #6", with its top-left corner at 10,0
A little squiggle coming down from the top of the component, a little way in from the left edge
Ans : b.

What does the following paint( ) method draw?
Public void paint(Graphics g) {
g.drawString("question #6",10,0);
}
A circle at (100, 100) with radius of 44
A circle at (100, 44) with radius of 100
A circle at (100, 44) with radius of 44
The code does not compile
Ans : d.
8)What is relationship between the Canvas class and the Graphics class?
Ans : A Canvas object provides access to a Graphics object via its paint( ) method.
What are the Component subclasses that support painting.
Ans : The Canvas, Frame, Panel and Applet classes support painting.
What is the difference between the paint( ) and repaint( ) method?
Ans : The paint( ) method supports painting via a Graphics object. The repaint( ) method is used to cause paint( ) to be invoked by the AWT painting method.
What is the difference between the Font and FontMetrics classes?
Ans : The FontMetrics class is used to define implementation-specific properties, such as ascent

and descent, of a Font object.
Which of the following are passed as an argument to the paint( ) method?
A Canvas object
A Graphics object
An Image object
A paint object
Ans : b.
Which of the following methods are invoked by the AWT to support paint and repaint operations?
paint( )
repaint( )
draw( )
redraw( )
Ans : a.
Which of the following classes have a paint( ) method?
Canvas
Image
Frame
Graphics
Ans : a and c.
Which of the following are methods of the Graphics class?
drawRect( )
drawImage( )
drawPoint( )
drawString( )
Ans : a, b and d.
Which Font attributes are available through the FontMetrics class?
ascent
leading
case
height
Ans : a, b and d.
Which of the following are true?
The AWT automatically causes a window to be repainted when a portion of a window has been minimized and then maximized.
The AWT automatically causes a window to be repainted when a portion of a window has been covered and then uncovered.
The AWT automatically causes a window to be repainted when application data is changed.
The AWT does not support repainting operations.
Ans : a and b.
Which method is used to size a graphics object to fit the current size of the window?
Ans : getSize( ) method.
What are the methods to be used to set foreground and background colors?
Ans : setForeground( ) and setBackground( ) methods.
19) You have created a simple Frame and overridden the paint method as follows

```
public void paint(Graphics g){

 g.drawString("Dolly",50,10);

}
```

What will be the result when you attempt to compile and run the program?
The string "Dolly" will be displayed at the centre of the frame
b) An error at compilation complaining at the signature of the paint method
c) The lower part of the word Dolly will be seen at the top of the form, with the top hidden.
d) The string "Dolly" will be shown at the bottom of the form
Ans : c.
20) Where g is a graphics instance what will the following code draw on the screen.

g.fillArc(45,90,50,50,90,180);
a) An arc bounded by a box of height 45, width 90 with a centre point of 50,50, starting at an angle of 90 degrees traversing through 180 degrees counter clockwise.
b) An arc bounded by a box of height 50, width 50, with a centre point of 45,90 starting at an angle of 90 degrees traversing through 180 degrees clockwise.
c) An arc bounded by a box of height 50, width 50, with a top left at coordinates of 45, 90, starting at 90 degrees and traversing through 180 degrees counter clockwise.
d) An arc starting at 45 degrees, traversing through 90 degrees clockwise bounded by a box of height 50, width 50 with a centre point of 90, 180.
Ans : c.

21) Given the following code

```
import java.awt.*;
public class SetF extends Frame{
public static void main(String argv[]){
SetF s = new SetF();
s.setSize(300,200);
s.setVisible(true);
}
}
```

How could you set the frame surface color to pink
a)s.setBackground(Color.pink);
b)s.setColor(PINK);
c)s.Background(pink);
d)s.color=Color.pink
Ans : a.

AWT: CONTROLS, LAYOUT MANAGERS AND MENUS

What is meant by Controls and what are different types of controls?
Ans : Controls are componenets that allow a user to interact with your application.
The AWT supports the following types of controls:
Labels
Push buttons
Check boxes
Choice lists
Lists
Scroll bars
Text components
These controls are subclasses of Component.

You want to construct a text area that is 80 character-widths wide and 10 character-heights tall. What code do you use?
new TextArea(80, 10)
new TextArea(10, 80)
Ans: b.

A text field has a variable-width font. It is constructed by calling new TextField("iiiii"). What happens if you change the contents of the text field to "wwwww"? (Bear in mind that is one of the narrowest characters, and w is one of the widest.)
The text field becomes wider.
The text field becomes narrower.
The text field stays the same width; to see the entire contents you will have to scroll by using the ß and à keys.
The text field stays the same width; to see the entire contents you will have to scroll by using the text field's horizontal scroll bar.
Ans : c.

The CheckboxGroup class is a subclass of the Component class.
True
False
Ans : b.
5) What are the immediate super classes of the following classes?
a) Container class
b) MenuComponent class
c) Dialog class
d) Applet class
e) Menu class
Ans : a) Container - Component
b) MenuComponent - Object
c) Dialog - Window
d) Applet - Panel
e) Menu - MenuItem
6) What are the SubClass of Textcomponent Class?
Ans : TextField and TextArea
7) Which method of the component class is used to set the position and the size of a component?
Ans : setBounds()
8) Which TextComponent method is used to set a TextComponent to the read-only state?
Ans : setEditable()
9) How can the Checkbox class be used to create a radio button?
Ans : By associating Checkbox objects with a CheckboxGroup.
10) What Checkbox method allows you to tell if a Checkbox is checked?
Ans : getState()
11) Which Component method is used to access a component's immediate Container?
getVisible()
getImmediate
getParent()
getContainer
Ans : c.
12) What methods are used to get and set the text label displayed by a Button object?
Ans : getLabel( ) and setLabel( )
13) What is the difference between a Choice and a List?
Ans : A Choice is displayed in a compact form that requires you to pull it down to see the list of available choices. Only one item may be selected from a Choice.
A List may be displayed in such a way that several List items are visible. A List supports the selection of one or more List items.
14) Which Container method is used to cause a container to be laid out and redisplayed?
Ans : validate( )
15) What is the difference between a Scollbar and a Scrollpane?
Ans : A Scrollbar is a Component, but not a Container.
A Scrollpane is a Container and handles its own events and performs its own scrolling.
16) Which Component subclass is used for drawing and painting?
Ans : Canvas.
17) Which of the following are direct or indirect subclasses of Component?
Button
Label
CheckboxMenuItem
Toolbar
Frame
Ans : a, b and e.
18) Which of the following are direct or indirect subclasses of Container?
Frame
TextArea

MenuBar
FileDialog
Applet
Ans : a,d and e.
19) Which method is used to set the text of a Label object?
setText( )
setLabel( )
setTextLabel( )
setLabelText( )
Ans : a.
20) Which constructor creates a TextArea with 10 rows and 20 columns?
new TextArea(10, 20)
new TextArea(20, 10)
new TextArea(new Rows(10), new columns(20))
new TextArea(200)
Ans : a.
(Usage is TextArea(rows, columns)
21) Which of the following creates a List with 5 visible items and multiple selection enabled?
new List(5, true)
new List(true, 5)
new List(5, false)
new List(false,5)
Ans : a.
[Usage is List(rows, multipleMode)]
22) Which are true about the Container class?
The validate( ) method is used to cause a Container to be laid out and redisplayed.
The add( ) method is used to add a Component to a Container.
The getBorder( ) method returns information about a Container's insets.
The getComponent( ) method is used to access a Component that is contained in a Container.
Ans : a, b and d.
23) Suppose a Panel is added to a Frame and a Button is added to the Panel. If the Frame's font is set to 12-point TimesRoman, the Panel's font is set to 10-point TimesRoman, and the Button's font is not set, what font will be used to dispaly the Button's label?
12-point TimesRoman
11-point TimesRoman
10-point TimesRoman
9-point TimesRoman
Ans : c.
A Frame's background color is set to Color.Yellow, and a Button's background color is to Color.Blue. Suppose the Button is added to a Panel, which is added to the Frame. What background color will be used with the Panel?
Colr.Yellow
Color.Blue
Color.Green
Color.White
Ans : a.
25) Which method will cause a Frame to be displayed?
show( )
setVisible( )
display( )
displayFrame( )
Ans : a and b.
26) All the componenet classes and container classes are derived from _____ class.
Ans : Object.
27) Which method of the container class can be used to add components to a Panel.
Ans : add ( ) method.

28) What are the subclasses of the Container class?
Ans : The Container class has three major subclasses. They are :
Window
Panel
ScrollPane
29) The Choice component allows multiple selection.
True.
False.
Ans : b.
30) The List component does not generate any events.
True.
False.
Ans : b.
31) Which components are used to get text input from the user.
Ans : TextField and TextArea.
32) Which object is needed to group Checkboxes to make them exclusive?
Ans : CheckboxGroup.
33) Which of the following components allow multiple selections?
Non-exclusive Checkboxes.
Radio buttons.
Choice.
List.
Ans : a and d.
34) What are the types of Checkboxes and what is the difference between them?
Ans : Java supports two types of Checkboxes. They are : Exclusive and Non-exclusive.
In case of exclusive Checkboxes, only one among a group of items can be selected at a time. I f an item from the group is selected, the checkbox currently checked is deselected and the new selection is highlighted. The exclusive Checkboxes are also called as Radio buttons.
The non-exclusive checkboxes are not grouped together and each one can be selected independent of the other.
35) What is a Layout Manager and what are the different Layout Managers available in java.awt and what is the default Layout manager for the panal and the panal subclasses?
Ans: A layout Manager is an object that is used to organize components in a container.
The different layouts available in java.awt are :
FlowLayout, BorderLayout, CardLayout, GridLayout and GridBag Layout.
The default Layout Manager of Panal and Panal sub classes is FlowLayout".
36) Can I exert control over the size and placement of components in my interface?
Ans : Yes.
myPanal.setLayout(null);
myPanal.setbounds(20,20,200,200);
37) Can I add the same component to more than one container?
Ans : No. Adding a component to a container automatically removes it from any previous parent(container).
38) How do I specify where a window is to be placed?
Ans : Use setBounds, setSize, or setLocation methods to implement this.
setBounds(int x, int y, int width, int height)
setBounds(Rectangle r)
setSize(int width, int height)
setSize(Dimension d)
setLocation(int x, int y)
setLocation(Point p)

39) How can we create a borderless window?
Ans : Create an instance of the Window class, give it a size, and show it on the screen.
eg. Frame aFrame = ...
Window aWindow = new Window(aFrame);

```
aWindow.setLayout(new FlowLayout());
aWindow.add(new Button("Press Me"));
aWindow.getBounds(50,50,200,200);
aWindow.show();
```

40) Can I create a non-resizable windows? If so, how?
Ans: Yes. By using setResizable() method in class Frame.
41) What is the default Layout Manager for the Window and Window subclasses (Frame,Dialog)?
Ans : BorderLayout().
42) How are the elements of different layouts organized?
Ans : FlowLayout : The elements of a FlowLayout are organized in a top to bottom, left to right fashion.
BorderLayout : The elements of a BorderLayout are organized at the borders (North, South, East and West) and the center of a container.
CardLayout : The elements of a CardLayout are stacked, one on top of the other, like a deck of cards.
GridLayout : The elements of a GridLayout are of equal size and are laid out using the square of a grid.
GridBagLayout : The elements of a GridBagLayout are organized according to a grid.However, the elements are of different sizes and may occupy more than one row or column of the grid. In addition, the rows and columns may have different sizes.
43) Which containers use a BorderLayout as their default layout?
Ans : The Window, Frame and Dialog classes use a BorderLayout as their default layout.

44) Which containers use a FlowLayout as their default layout?
Ans : The Panel and the Applet classes use the FlowLayout as their default layout.
45) What is the preferred size of a component?
Ans : The preferred size of a component size that will allow the component to display normally.
46) Which method is method to set the layout of a container?
startLayout( )
initLayout( )
layoutContainer( )
setLayout( )
Ans : d.
47) Which method returns the preferred size of a component?
getPreferredSize( )
getPreferred( )
getRequiredSize( )
getLayout( )
Ans : a.

48) Which layout should you use to organize the components of a container in a tabular form?
CardLayout
BorederLayout
FlowLayout
GridLayout
Ans : d.
An application has a frame that uses a Border layout manager. Why is it probably not a good idea to put a vertical scroll bar at North in the frame?
The scroll bar's height would be its preferred height, which is not likely to be enough.
The scroll bar's width would be the entire width of the frame, which would be much wider than necessary.

Both a and b.
Neither a nor b. There is no problem with the layout as described.
Ans : c.
What is the default layouts for a applet, a frame and a panel?
Ans : For an applet and a panel, Flow layout is the default layout, whereas Border layout is default layout for a frame.
If a frame uses a Grid layout manager and does not contain any panels, then all the components within the frame are the same width and height.
True
False.
Ans : a.
If a frame uses its default layout manager and does not contain any panels, then all the components within the frame are the same width and height.
True
False.
Ans : b.
With a Border layout manager, the component at Center gets all the space that is left over, after the components at North and South have been considered.
True
False
Ans : b.
An Applet has its Layout Manager set to the default of FlowLayout. What code would be the correct to change to another Layout Manager?
setLayoutManager(new GridLayout());
setLayout(new GridLayout(2,2));
c) setGridLayout(2,2,))
d setBorderLayout();
Ans : b.
55) How do you indicate where a component will be positioned using Flowlayout?
a) North, South,East,West
b) Assign a row/column grid reference
c) Pass a X/Y percentage parameter to the add method
d) Do nothing, the FlowLayout will position the component
Ans :d.

56) How do you change the current layout manager for a container?
a) Use the setLayout method
b) Once created you cannot change the current layout manager of a component
c) Use the setLayoutManager method
d) Use the updateLayout method
Ans :a.
57)When using the GridBagLayout manager, each new component requires a new instance of the GridBagConstraints class. Is this statement true or false?
a) true
b) false
Ans : b.
58) Which of the following statements are true?
a)The default layout manager for an Applet is FlowLayout
b) The default layout manager for an application is FlowLayout
c) A layout manager must be assigned to an Applet before the setSize method is called
d) The FlowLayout manager attempts to honor the preferred size of any components
Ans : a and d.
59) Which method does display the messages whenever there is an item selection or deselection of the CheckboxMenuItem menu?

Ans : itemStateChanged method.
60) Which is a dual state menu item?
Ans : CheckboxMenuItem.
61) Which method can be used to enable/diable a checkbox menu item?
Ans : setState(boolean).
Which of the following may a menu contain?
A separator
A check box
A menu
A button
A panel
Ans : a and c.
Which of the following may contain a menu bar?
A panel
A frame
An applet
A menu bar
A menu
Ans : b
64) What is the difference between a MenuItem and a CheckboxMenuItem?
Ans : The CheckboxMenuItem class extends the MenuItem class to support a menu item
that may be checked or unchecked.

65) Which of the following are true?
A Dialog can have a MenuBar.
MenuItem extends Menu.
A MenuItem can be added to a Menu.
A Menu can be added to a Menu.
Ans : c and d.

Which colour is used to indicate instance methods in the standard "javadoc" format documentation:
1) blue
2) red
3) purple
4) orange
Answer : 2
explain
In JDK 1.1 the variabels, methods and constructors are colour coded to simplifytheir identification.
endExplain
What is the correct ordering for the import, class and package declarations when found in a single file?
1) package, import, class
2) class, import, package
3) import, package, class
4) package, class, import
Answer : 1
explain
This is my explanation for question 2
endExplain
Which methods can be legally applied to a string object?
(Multiple)
1) equals(String)
2) equals(Object)
3) trim()
4) round()
5) toString()

Answer : 1,2,3,5
What is the parameter specification for the public static void main method?
(multiple)
1) String args []
2) String [] args
3) Strings args []
4) String args
Answer : 1,2
What does the zeroth element of the string array passed to the public static void main method contain?
(multiple)
1) The name of the program
2) The number of arguments
3) The first argument if one is present
Answer : 3
Which of the following are Java keywords?
(multiple)
1) goto
2) malloc
3) extends
4) FALSE
Answer : 3
What will be the result of compiling the following code:

```
public class Test {
public static void main (String args []) {
int age;
age = age + 1;
System.out.println("The age is " + age);
}
}
```

1) Compiles and runs with no output
2) Compiles and runs printing out The age is 1
3) Compiles but generates a runtime error
4) Does not compile
5) Compiles but generates a compile time error
Answer : 4
Which of these is the correct format to use to create the literal char value a?
(multiple)
1) 'a'
2) "a"
3) new Character(a)
4) \000a
Answer : 1
What is the legal range of a byte integral type?
1) 0 - 65, 535
2) (-128) - 127
3) (-32,768) - 32,767
4) (-256) - 255
Answer : 2
Which of the following is illegal:
1) int i = 32;
2) float f = 45.0;
3) double d = 45.0;
Answer 2
What will be the result of compiling the following code:

```
public class Test {
static int age;
```

```
public static void main (String args []) {
age = age + 1;
System.out.println("The age is " + age);
}
}
```

1) Compiles and runs with no output
2) Compiles and runs printing out The age is 1
3) Compiles but generates a runtime error
4) Does not compile
5) Compiles but generates a compile time error

Answer : 2

Which of the following are correct?
(multiple)
1) 128 >> 1 gives 64
2) 128 >>> 1 gives 64
3) 128 >> 1 gives -64
4) 128 >>> 1 gives -64

Answer : 1

Which of the following return true?
(multiple)
1) "john" == new String("john")
2) "john".equals("john")
3) "john" = "john"
4) "john".equals(new Button("john"))

Answer : 2

Which of the following do not lead to a runtime error?
(multiple)
1) "john" + " was " + " here"
2) "john" + 3
3) 3 + 5
4) 5 + 5.5

answer 1,2,3,4

Which of the following are so called "short circuit" logical operators?
(multiple)
1) &
2) ||
3) &&
4) |

Answer : 2,3

Which of the following are acceptable?
(multiple)
1) Object o = new Button("A");
2) Boolean flag = true;
3) Panel p = new Frame();
4) Frame f = new Panel();
5) Panel p = new Applet();

Answer : 1,5

What is the result of compiling and running the following code:

```
public class Test {
static int total = 10;
public static void main (String args []) {
new Test();
}
public Test () {
System.out.println("In test");
System.out.println(this);
```

```
int temp = this.total;
if (temp > 5) {
System.out.println(temp);
}
}
}
```

(multiple)
1) The class will not compile
2) The compiler reports and error at line 2
3) The compiler reports an error at line 9
4) The value 10 is one of the elements printed to the standard output
5) The class compiles but generates a runtime error
Answer : 4
Which of the following is correct:
1) String temp [] = new String {"j" "a" "z"};
2) String temp [] = { "j " " b" "c"};
3) String temp = {"a", "b", "c"};
4) String temp [] = {"a", "b", "c"};
Answer 4
What is the correct declaration of an abstract method that is intended to be public:
1) public abstract void add();
2) public abstract void add() {}
3) public abstract add();
4) public virtual add();
Answer : 1
Under what situations do you obtain a default constructor?
1) When you define any class
2) When the class has no other constructors
3) When you define at least one constructor
Answer : 2
Which of the following can be used to define a constructor for this class, given the following code:

```
public class Test {
...
}
```

1) public void Test() {...}
2) public Test() {...}
3) public static Test() {...}
4) public static void Test() {...}
Answer : 2
Which of the following are acceptable to the Java compiler:
(multiple)
1) if (2 == 3) System.out.println("Hi");
2) if (2 = 3) System.out.println("Hi");
3) if (true) System.out.println("Hi");
4) if (2 != 3) System.out.println("Hi");
5) if (aString.equals("hello")) System.out.println("Hi");
Answer : 1,3,4,5
Assuming a method contains code which may raise an Exception (but not a RuntimeException), what is the correct way for a method to indicate that it expects the caller to handle that exception:
1) throw Exception
2) throws Exception
3) new Exception
4) Don't need to specify anything
Answer : 2
What is the result of executing the following code, using the parameters 4 and 0:

```
public void divide(int a, int b) {
```

```
try {
int c = a / b;
} catch (Exception e) {
System.out.print("Exception ");
} finally {
System.out.println("Finally");
}
```

1) Prints out: Exception Finally
2) Prints out: Finally
3) Prints out: Exception
4) No output

Answer : 1

Which of the following is a legal return type of a method overloading the following method:

```
public void add(int a) {...}
```

1) void
2) int
3) Can be anything

Answer : 3

Which of the following statements is correct for a method which is overriding the following method:

```
public void add(int a) {...}
```

1) the overriding method must return void
2) the overriding method must return int
3) the overriding method can return whatever it likes

Answer : 1

Given the following classes defined in separate files, what will be the effect of compiling and running this class Test?

```
class Vehicle {
public void drive() {
System.out.println("Vehicle: drive");
}
}
class Car extends Vehicle {
public void drive() {
System.out.println("Car: drive");
}
}
public class Test {
public static void main (String args []) {
Vehicle v;
Car c;
v = new Vehicle();
c = new Car();
v.drive();
c.drive();
v = c;
v.drive();
}
}
```

1) Generates a Compiler error on the statement v= c;
2) Generates runtime error on the statement v= c;
3) Prints out:
Vehicle: drive
Car: drive
Car: drive
4) Prints out:
Vehicle: drive

Car: drive
Vehicle: drive
Answer : 3
Where in a constructor, can you place a call to a constructor defined in the super class?
1) Anywhere
2) The first statement in the constructor
3) The last statement in the constructor
4) You can't call super in a constructor
Answer : 2
Which variables can an inner class access from the class which encapsulates it?
(multiple)
1) All static variables
2) All final variables
3) All instance variables
4) Only final instance variables
5) Only final static variables
Answer : 1,2,3
What class must an inner class extend:
1) The top level class
2) The Object class
3) Any class or interface
4) It must extend an interface
Answer 3
In the following code, which is the earliest statement, where the object originally held in e, may be garbage collected:

```
1. public class Test {
2. public static void main (String args []) {
3. Employee e = new Employee("Bob", 48);
4. e.calculatePay();
5. System.out.println(e.printDetails());
6. e = null;
7. e = new Employee("Denise", 36);
8. e.calculatePay();
9. System.out.println(e.printDetails());
10. }
11. }
```

1) Line 10
2) Line 11
3) Line 7
4) Line 8
5) Never
Answer : 3
What is the name of the interface that can be used to define a class that can execute within its own thread?
1) Runnable
2) Run
3) Threadable
4) Thread
5) Executable
Answer : 1
What is the name of the method used to schedule a thread for execution?
1) init();
2) start();
3) run();
4) resume();
5) sleep();

Answer : 2
Which methods may cause a thread to stop executing?
(multiple)
1) sleep();
2) stop();
3) yield();
4) wait();
5) notify();
6) notifyAll()
7) synchronized()
Answer : 1,2,3,4
Which of the following would create a text field able to display 10 characters (assuming a fixed size font) displaying the initial string "hello":
1) new TextField("hello", 10);
2) new TextField("hello");
3) new textField(10);
4) new TextField();
Answer : 1
Which of the following methods are defined on the Graphics class:
(multiple)
1) drawLine(int, int, int, int)
2) drawImage(Image, int, int, ImageObserver)
3) drawString(String, int, int)
4) add(Component);
5) setVisible(boolean);
6) setLayout(Object);
Answer : 1,2,3
Which of the following layout managers honours the preferred size of a component:
(multiple)
1) CardLayout
2) FlowLayout
3) BorderLayout
4) GridLayout
Answer : 2
Given the following code what is the effect of a being 5:

```
public class Test {
public void add(int a) {
loop: for (int i = 1; i < 3; i++){
for (int j = 1; j < 3; j++) {
if (a == 5) {
break loop;
}
System.out.println(i * j);
}
}
}
}
```

1) Generate a runtime error
2) Throw an ArrayIndexOutOfBoundsException
3) Print the values: 1, 2, 2, 4
4) Produces no output
Answer : 4
What is the effect of issuing a wait() method on an object
1) If a notify() method has already been sent to that object then it has no effect
2) The object issuing the call to wait() will halt until another object sends a notify() or notifyAll() method

3) An exception will be raised
4) The object issuing the call to wait() will be automatically synchronized with any other objects using the receiving object.
Answer : 2
The layout of a container can be altered using which of the following methods:
(multiple)
1) setLayout(aLayoutManager);
2) addLayout(aLayoutManager);
3) layout(aLayoutManager);
4) setLayoutManager(aLayoutManager);
Answer : 1
Using a FlowLayout manager, which is the correct way to add elements to a container:
1) add(component);
2) add("Center", component);
3) add(x, y, component);
4) set(component);
Answer : 1
Given that a Button can generate an ActionEvent which listener would you expect to have to implement, in a class which would handle this event?
1) FocusListener
2) ComponentListener
3) WindowListener
4) ActionListener
5) ItemListener
Answer : 4
Which of the following, are valid return types, for listener methods:
1) boolean
2) the type of event handled
3) void
4) Component
Answer : 3
Assuming we have a class which implements the ActionListener interface, which method should be used to register this with a Button?
1) addListener(*);
2) addActionListener(*);
3) addButtonListener(*);
4) setListener(*);
Answer : 2
In order to cause the paint(Graphics) method to execute, which of the following is the most appropriate method to call:
1) paint()
2) repaint()
3) paint(Graphics)
4) update(Graphics)
5) None - you should never cause paint(Graphics) to execute
Answer : 2
Which of the following illustrates the correct way to pass a parameter into an applet:
1) <applet code=Test.class age=33 width=100 height=100>
2) <param name=age value=33>
3) <applet code=Test.class name=age value=33 width=100 height=100>
4) <applet Test 33>
Answer : 2
Which of the following correctly illustrate how an InputStreamReader can be created:
(multiple)
1) new InputStreamReader(new FileInputStream("data"));
2) new InputStreamReader(new FileReader("data"));

3) new InputStreamReader(new BufferedReader("data"));
4) new InputStreamReader("data");
5) new InputStreamReader(System.in);
Answer : 1,5
What is the permanent effect on the file system of writing data to a new FileWriter("report"), given the file report already exists?
1) The data is appended to the file
2) The file is replaced with a new file
3) An exception is raised as the file already exists
4) The data is written to random locations within the file
Answer : 2
What is the effect of adding the sixth element to a vector created in the following manner:
new Vector(5, 10);
1) An IndexOutOfBounds exception is raised.
2) The vector grows in size to a capacity of 10 elements
3) The vector grows in size to a capacity of 15 elements
4) Nothing, the vector will have grown when the fifth element was added
Answer : 3
What is the result of executing the following code when the value of x is 2:

```
switch (x) {
case 1:
System.out.println(1);
case 2:
case 3:
System.out.println(3);
case 4:
System.out.println(4);
}
```

1) Nothing is printed out
2) The value 3 is printed out
3) The values 3 and 4 are printed out
4) The values 1, 3 and 4 are printed out
Answer : 3
What is the result of compiling and running the Second class?
Consider the following example:

```
class First {
public First (String s) {
System.out.println(s);
}
}
public class Second extends First {
public static void main(String args []) {
new Second();
}
}
```

1) Nothing happens
2) A string is printed to the standard out
3) An instance of the class First is generated
4) An instance of the class Second is created
5) An exception is raised at runtime stating that there is no null parameter constructor in class First.
6) The class second will not compile as there is no null parameter constructor in the class First
Answer : 6
What is the result of executing the following fragment of code:

```
boolean flag = false;
if (flag = true) {
System.out.println("true");
```

```
} else {
System.out.println("false");
}
```
1) true is printed to standard out
2) false is printed to standard out
3) An exception is raised
4) Nothing happens
Answer : 1
Consider the following classes. What is the result of compiling and running this class?
```
public class Test {
public static void test() {
this.print();
}
public static void print() {
System.out.println("Test");
}
public static void main(String args []) {
test();
}
}
```
(multiple)
1) The string Test is printed to the standard out.
2) A runtime exception is raised stating that an object has not been created.
3) Nothing is printed to the standard output.
4) An exception is raised stating that the method test cannot be found.
5) An exception is raised stating that the variable this can only be used within an instance.
6) The class fails to compile stating that the variable this is undefined.
Answer : 6
Examine the following class definition:
```
public class Test {
public static void test() {
print();
}
public static void print() {
System.out.println("Test");
}
public void print() {
System.out.println("Another Test");
}
}
```
What is the result of compiling this class:
1) A successful compilation.
2) A warning stating that the class has no main method.
3) An error stating that there is a duplicated method.
4) An error stating that the method test() will call one or other of the print() methods.
Answer : 3
What is the result of compiling and executing the following Java class:
```
public class ThreadTest extends Thread {
public void run() {
System.out.println("In run");
suspend();
resume();
System.out.println("Leaving run");
}
public static void main(String args []) {
(new ThreadTest()).start();
```

```
}
}
```

1) Compilation will fail in the method main.
2) Compilation will fail in the method run.
3) A warning will be generated for method run.
4) The string "In run" will be printed to standard out.
5) Both strings will be printed to standard out.
6) Nothing will happen.

Answer : 4

Given the following sequence of Java statements, Which of the following options are true:

```
1. StringBuffer sb = new StringBuffer("abc");
2. String s = new String("abc");
3. sb.append("def");
4. s.append("def");
5. sb.insert(1, "zzz");
6. s.concat(sb);
7. s.trim();
```

(multiple)

1) The compiler would generate an error for line 1.
2) The compiler would generate an error for line 2.
3) The compiler would generate an error for line 3.
4) The compiler would generate an error for line 4.
5) The compiler would generate an error for line 5.
6) The compiler would generate an error for line 6.
7) The compiler would generate an error for line 7.

Answer : 4,6

What is the result of executing the following Java class:

```
import java.awt.*;
public class FrameTest extends Frame {
public FrameTest() {
add (new Button("First"));
add (new Button("Second"));
add (new Button("Third"));
pack();
setVisible(true);
}
public static void main(String args []) {
new FrameTest();
}
}
```

1) Nothing happens.
2) Three buttons are displayed across a window.
3) A runtime exception is generated (no layout manager specified).
4) Only the "first" button is displayed.
5) Only the "second" button is displayed.
6) Only the "third" button is displayed.

Answer : 6

Consider the following tags and attributes of tags, which can be used with the <AAPLET> and </APPLET> tags?

1. CODEBASE
2. ALT
3. NAME
4. CLASS
5. JAVAC
6. HORIZONTALSPACE
7. VERTICALSPACE

8. WIDTH
9. PARAM
10. JAR
(multiple)
1) line 1, 2, 3
2) line 2, 5, 6, 7
3) line 3, 4, 5
4) line 8, 9, 10
5) line 8, 9
Answer : 1,5
Which of the following is a legal way to construct a RandomAccessFile:
1) RandomAccessFile("data", "r");
2) RandomAccessFile("r", "data");
3) RandomAccessFile("data", "read");
4) RandomAccessFile("read", "data");
Answer : 1
Carefully examine the following code, When will the string "Hi there" be printed?

```
public class StaticTest {
static {
System.out.println("Hi there");
}
public void print() {
System.out.println("Hello");
}
public static void main(String args []) {
StaticTest st1 = new StaticTest();
st1.print();
StaticTest st2 = new StaticTest();
st2.print();
}
}
```

1) Never.
2) Each time a new instance is created.
3) Once when the class is first loaded into the Java virtual machine.
4) Only when the static method is called explicitly.
Answer : 3
What is the result of the following program:

```
public class Test {
public static void main (String args []) {
boolean a = false;
if (a = true)
System.out.println("Hello");
else
System.out.println("Goodbye");
}
}
```

1) Program produces no output but terminates correctly.
2) Program does not terminate.
3) Prints out "Hello"
4) Prints out "Goodbye"
Answer : 3
Examine the following code, it includes an inner class, what is the result:

```
public final class Test4 {
class Inner {
void test() {
if (Test4.this.flag); {
```

```
sample();
}
}
}
private boolean flag = true;
public void sample() {
System.out.println("Sample");
}
public Test4() {
(new Inner()).test();
}
public static void main(String args []) {
new Test4();
}
}
```

1) Prints out "Sample"
2) Program produces no output but terminates correctly.
3) Program does not terminate.
4) The program will not compile

Answer : 1

Carefully examine the following class:

```
public class Test5 {
public static void main (String args []) {
/* This is the start of a comment
if (true) {
Test5 = new test5();
System.out.println("Done the test");
}
/* This is another comment */
System.out.println ("The end");
}
}
```

1) Prints out "Done the test" and nothing else.
2) Program produces no output but terminates correctly.
3) Program does not terminate.
4) The program will not compile.
5) The program generates a runtime exception.
6) The program prints out "The end" and nothing else.
7) The program prints out "Done the test" and "The end"

Answer : 6

What is the result of compiling and running the following applet:

```
import java.applet.Applet;
import java.awt.*;
public class Sample extends Applet {
private String text = "Hello World";
public void init() {
add(new Label(text));
}
public Sample (String string) {
text = string;
}
}
```

It is accessed form the following HTML page:

```
<html>
<title>Sample Applet</title>
<body>
```

```
<applet code="Sample.class" width=200 height=200></applet>
</body>
</html>
```

1) Prints "Hello World".
2) Generates a runtime error.
3) Does nothing.
4) Generates a compile time error.

Answer : 2

What is the effect of compiling and (if possible) running this class:

```
public class Calc {
public static void main (String args []) {
int total = 0;
for (int i = 0, j = 10; total > 30; ++i, --j) {
System.out.println(" i = " + i + " : j = " + j);
total += (i + j);
}
System.out.println("Total " + total);
}
}
```

1) Produce a runtime error
2) Produce a compile time error
3) Print out "Total 0"
4) Generate the following as output:

i = 0 : j = 10
i = 1 : j = 9
i = 2 : j = 8
Total 30

Answer : 3

Utility Package

1) What is the Vector class?

ANSWER : The Vector class provides the capability to implement a growable array of objects.

2) What is the Set interface?

ANSWER : The Set interface provides methods for accessing the elements of a finite mathematical set.Sets do not allow duplicate elements.

3) What is Dictionary class?

ANSWER : The Dictionary class is the abstarct super class of Hashtable and Properties class.Dictionary provides the abstarct functions used to store and retrieve objects by key-value.This class allows any object to be used as a key or value.

4) What is the Hashtable class?

ANSWER : The Hashtable class implements a hash table data structure. A hash table indexes and stores objects in a dictionary using hash codes as the objects' keys. Hash codes are integer values that identify objects.

5) What is the Properties class?

Answer : The properties class is a subclass of Hashtable that can be read from or written to a stream.It also provides the capability to specify a set of default values to be used if a specified key is not found in the table. We have two methods load() and save().

6) What changes are needed to make the following prg to compile?

```
import java.util.*;
class Ques{
public static void main (String args[]) {
String s1 = "abc";
String s2 = "def";
Vector v = new Vector();
v.add(s1);
v.add(s2);
String s3 = v.elementAt(0) + v.elementAt(1);
System.out.println(s3);
}
}
```

ANSWER : Declare Ques as public B) Cast v.elementAt(0) to a String
C) Cast v.elementAt(1) to an Object. D) Import java.lang
ANSWER : B) Cast v.elementAt(0) to a String

8) What is the output of the prg.

```
import java.util.*;
class Ques{
public static void main (String args[]) {
String s1 = "abc";
String s2 = "def";
Stack stack = new Stack();
stack.push(s1);
stack.push(s2);
try{
String s3 = (String) stack.pop() + (String) stack.pop() ;
System.out.println(s3);
}catch (EmptyStackException ex){}
}
}
```

ANSWER : abcdef B) defabc C) abcabc D) defdef
ANSWER : B) defabc

9) Which of the following may have duplicate elements?

ANSWER : Collection B) List C) Map D) Set
ANSWER : A and B Neither a Map nor a Set may have duplicate elements.

10) Can null value be added to a List?

ANSWER : Yes.A Null value may be added to any List.

11) What is the output of the following prg.

```
import java.util.*;
class Ques{
public static void main (String args[]) {
HashSet set = new HashSet();
String s1 = "abc";
String s2 = "def";
String s3 = "";
set.add(s1);
set.add(s2);
```

```
set.add(s1);
set.add(s2);
Iterator i = set.iterator();
while(i.hasNext())
{
s3 += (String) i.next();
}
System.out.println(s3);
}
}
```

A) abcdefabcdef B) defabcdefabc C) fedcbafedcba D) defabc

ANSWER : D) defabc. Sets may not have duplicate elements.

12) Which of the following java.util classes support internationalization?

A) Locale B) ResourceBundle C) Country D) Language

ANSWER : A and B . Country and Language are not java.util classes.

13) What is the ResourceBundle?

The ResourceBundle class also supports internationalization.
ResourceBundle subclasses are used to store locale-specific resources that can be loaded by a program to tailor the program's appearence to the paticular locale in which it is being run. Resource Bundles provide the capability to isolate a program's locale-specific resources in a standard and modular manner.

14) How are Observer Interface and Observable class, in java.util package, used?

ANSWER : Objects that subclass the Observable class maintain a list of Observers. When an Observable object is updated it invokes the update() method of each of its observers to notify the observers that it has changed state. The Observer interface is implemented by objects that observe Observable objects.

15) Which java.util classes and interfaces support event handling?

ANSWER : The EventObject class and the EventListener interface support event processing.

16) Does java provide standard iterator functions for inspecting a collection of objects?

ANSWER : The Enumeration interface in the java.util package provides a framework for stepping once through a collection of objects. We have two methods in that interface.

```
public interface Enumeration {
boolean hasMoreElements();
Object nextElement();
}
```

17) The Math.random method is too limited for my needs- How can I generate random numbers more flexibly?

ANSWER : The random method in Math class provide quick, convienient access to random numbers, but more power and flexibility use the Random class in the java.util package.

```
double doubleval = Math.random();
```

The Random class provide methods returning float, int, double, and long values.

```
nextFloat() // type float; 0.0 <= value < 1.0
nextDouble() // type double; 0.0 <= value < 1.0
nextInt() // type int; Integer.MIN_VALUE <= value <= Integer.MAX_VALUE
nextLong() // type long; Long.MIN_VALUE <= value <= Long.MAX_VALUE
nextGaussian() // type double; has Gaussian("normal") distribution with mean 0.0 and standard deviation 1.0)
Eg. Random r = new Random();
float floatval = r.nextFloat();
```

18) How can we get all public methods of an object dynamically?

ANSWER : By using getMethods(). It return an array of method objects corresponding to the public methods of this class.
getFields() returns an array of Filed objects corresponding to the public Fields(variables) of this class.

getConstructors() returns an array of constructor objects corresponding to the public constructors of this class.

JDBC

1) What are the steps involved in establishing a connection?

ANSWER : This involves two steps: (1) loading the driver and (2) making the connection.

2) How can you load the drivers?

ANSWER : Loading the driver or drivers you want to use is very simple and involves just one line of code. If, for example, you want to use the JDBC-ODBC Bridge driver, the following code will load it:

Eg.

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Your driver documentation will give you the class name to use. For instance, if the class name is jdbc.DriverXYZ , you would load the driver with the following line of code:

Eg.

Class.forName("jdbc.DriverXYZ");

3) What Class.forName will do while loading drivers?

ANSWER : It is used to create an instance of a driver and register it with the DriverManager. When you have loaded a driver, it is available for making a connection with a DBMS.

4) How can you make the connection?

ANSWER : In establishing a connection is to have the appropriate driver connect to the DBMS. The following line of code illustrates the general idea:

Eg.

String url = "jdbc:odbc:Fred";

Connection con = DriverManager.getConnection(url, "Fernanda", "J8");

5) How can you create JDBC statements?

ANSWER : A Statement object is what sends your SQL statement to the DBMS. You simply create a Statement object and then execute it, supplying the appropriate execute method with the SQL statement you want to send. For a SELECT statement, the method to use is executeQuery. For statements that create or modify tables, the method to use is executeUpdate.

Eg.

It takes an instance of an active connection to create a Statement object. In the following example, we use our Connection object con to create the Statement object stmt :

Statement stmt = con.createStatement();

6) How can you retrieve data from the ResultSet?

ANSWER : Step 1.

JDBC returns results in a ResultSet object, so we need to declare an instance of the class ResultSet to hold our results. The following code demonstrates declaring the ResultSet object rs.

Eg.

ResultSet rs = stmt.executeQuery("SELECT COF_NAME, PRICE FROM COFFEES");

Step2.

String s = rs.getString("COF_NAME");

The method getString is invoked on the ResultSet object rs , so getString will retrieve (get) the value stored in the column COF_NAME in the current row of rs

7) What are the different types of Statements?

ANSWER : 1.Statement (use createStatement method) 2. Prepared Statement (Use prepareStatement method) and 3. Callable Statement (Use prepareCall)

8) How can you use PreparedStatement?

ANSWER : This special type of statement is derived from the more general class, Statement.If you want to execute a Statement object many times, it will normally reduce execution time to use a PreparedStatement object instead.

The advantage to this is that in most cases, this SQL statement will be sent to the DBMS right away, where it will be compiled. As a result, the PreparedStatement object contains not just an SQL statement, but an SQL statement that has been precompiled. This means that when the PreparedStatement is executed, the DBMS can just run the PreparedStatement 's SQL statement without having to compile it first.

Eg.
PreparedStatement updateSales = con.prepareStatement("UPDATE COFFEES SET SALES = ? WHERE COF_NAME LIKE ?");

9) What setAutoCommit does?

ANSWER : When a connection is created, it is in auto-commit mode. This means that each individual SQL statement is treated as a transaction and will be automatically committed right after it is executed. The way to allow two or more statements to be grouped into a transaction is to disable auto-commit mode

Eg.
con.setAutoCommit(false);

Once auto-commit mode is disabled, no SQL statements will be committed until you call the method commit explicitly.

Eg.
con.setAutoCommit(false);
PreparedStatement updateSales = con.prepareStatement(
"UPDATE COFFEES SET SALES = ? WHERE COF_NAME LIKE ?");
updateSales.setInt(1, 50);
updateSales.setString(2, "Colombian");
updateSales.executeUpdate();
PreparedStatement updateTotal = con.prepareStatement("UPDATE COFFEES SET TOTAL = TOTAL + ? WHERE COF_NAME LIKE ?");
updateTotal.setInt(1, 50);
updateTotal.setString(2, "Colombian");
updateTotal.executeUpdate();
con.commit();
con.setAutoCommit(true);

10) How to call a Strored Procedure from JDBC?

ANSWER : The first step is to create a CallableStatement object. As with Statement an and PreparedStatement objects, this is done with an open Connection
object. A CallableStatement object contains a call to a stored procedure;

Eg.
CallableStatement cs = con.prepareCall("{call SHOW_SUPPLIERS}");
ResultSet rs = cs.executeQuery();

11) How to Retrieve Warnings?

ANSWER : SQLWarning objects are a subclass of SQLException that deal with database access warnings. Warnings do not stop the execution of an application, as exceptions do; they simply alert the user that something did not happen as planned.

A warning can be reported on a Connection object, a Statement object (including PreparedStatement and CallableStatement objects), or a ResultSet object. Each of these classes has a getWarnings method, which you must invoke in order to see the first warning reported on the calling object

Eg.
SQLWarning warning = stmt.getWarnings();
if (warning != null) {
System.out.println("\n---Warning---\n");
while (warning != null) {
System.out.println("Message: " + warning.getMessage());
System.out.println("SQLState: " + warning.getSQLState());
System.out.print("Vendor error code: ");
System.out.println(warning.getErrorCode());
System.out.println("");
warning = warning.getNextWarning();
}
}

12) How can you Move the Cursor in Scrollable Result Sets ?

ANSWER : One of the new features in the JDBC 2.0 API is the ability to move a result set's cursor backward as well as forward. There are also methods that let you move the cursor to a particular row and check the position of the cursor.
Eg.
Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
ResultSet srs = stmt.executeQuery("SELECT COF_NAME, PRICE FROM COFFEES");
The first argument is one of three constants added to the ResultSet API to indicate the type of a ResultSet object: TYPE_FORWARD_ONLY, TYPE_SCROLL_INSENSITIVE , and TYPE_SCROLL_SENSITIVE .
The second argument is one of two ResultSet constants for specifying whether a result set is read-only or updatable: CONCUR_READ_ONLY and CONCUR_UPDATABLE . The point to remember here is that if you specify a type, you must also specify whether it is read-only or updatable. Also, you must specify the type first, and because both parameters are of type int , the compiler will not complain if you switch the order.
Specifying the constant TYPE_FORWARD_ONLY creates a nonscrollable result set, that is, one in which the cursor moves only forward. If you do not specify any constants for the type and updatability of a ResultSet object, you will automatically get one that is TYPE_FORWARD_ONLY and CONCUR_READ_ONLY

13) What's the difference between TYPE_SCROLL_INSENSITIVE , and TYPE_SCROLL_SENSITIVE?

ANSWER : You will get a scrollable ResultSet object if you specify one of these ResultSet constants.The difference between the two has to do with whether a result set reflects changes that are made to it while it is open and whether certain methods can be called to detect these changes. Generally speaking, a result set that is TYPE_SCROLL_INSENSITIVE does not reflect changes made while it is still open and one that is TYPE_SCROLL_SENSITIVE does. All three types of result sets will make changes visible if they are closed and then reopened
Eg.

```
Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
ResultSet srs = stmt.executeQuery("SELECT COF_NAME, PRICE FROM COFFEES");
srs.afterLast();
while (srs.previous()) {
String name = srs.getString("COF_NAME");
float price = srs.getFloat("PRICE");
System.out.println(name + " " + price);
}
```

14) How to Make Updates to Updatable Result Sets?

ANSWER : Another new feature in the JDBC 2.0 API is the ability to update rows in a result set using methods in the Java programming language rather than having to send an SQL command. But before you can take advantage of this capability, you need to create a ResultSet object that is updatable. In order to do this, you supply the ResultSet constant CONCUR_UPDATABLE to the createStatement method.
Eg.

```
Connection con = DriverManager.getConnection("jdbc:mySubprotocol:mySubName");
Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
ResultSet uprs = stmt.executeQuery("SELECT COF_NAME, PRICE FROM COFFEES");
```

Networking Concepts

1) The API doesn't list any constructors for InetAddress- How do I create an InetAddress instance?

ANSWER : In case of InetAddress the three methods getLocalHost, getByName, getByAllName can be used to create instances.

E.g.

```
InetAddress add1;
```

```
InetAddress add2;
try{
add1 = InetAddress.getByName("java.sun.com");
add2 = InetAddress.getByName("199.22.22.22");
}catch(UnknownHostException e){}
```

2) Is it possible to get the Local host IP?
ANSWER : Yes. Use InetAddress's getLocalHost method.
3) What's the Factory Method?
ANSWER : Factory methods are merely a convention whereby static methods in a class return an instance of that class. The InetAddress class has no visible constructors. To create an InetAddress object, you have to use one of the available factory methods. In InetAddress the three methods getLocalHost, getByName, getByAllName can be used to create instances of InetAddress.
4) What's the difference between TCP and UDP?
ANSWER : These two protocols differ in the way they carry out the action of communicating. A TCP protocol establishes a two way connection between a pair of computers, while the UDP protocol is a one-way message sender. The common analogy is that TCP is like making a phone call and carrying on a two-way communication, while UDP is like mailing a letter.
5) What is the Proxy Server?
ANSWER : A proxy server speaks the client side of a protocol to another server. This is often required when clients have certain restrictions on which servers they can connect to. And when several users are hitting a popular web site, a proxy server can get the contents of the web server's popular pages once, saving expensive internetwork transfers while providing faster access to those pages to the clients. Also, we can get multiple connections for a single server.
6) What are the seven layers of OSI model?
ANSWER : Application, Presentation, Session, Transport, Network, DataLink, Physical Layer.
What Transport Layer does?
ANSWER : It ensures that the mail gets to its destination. If a packet fails to get its destination, it handles the process of notifying the sender and requesting that another packet be sent.
8) What is DHCP?
ANSWER : Dynamic Host Configuration Protocol, a piece of the TCP/IP protocol suite that handles the automatic assignment of IP addresses to clients.
9) What is SMTP?
ANSWER : Simple Mail Transmission Protocol, the TCP/IP Standard for Internet mails. SMTP exchanges mail between servers; contrast this with POP, which transmits mail between a server and a client.
10) In OSI N/w architecture, the dialogue control and token management are responsibilities of...
Answer : Network b) Session c) Application d) DataLink
ANSWER : b) Session Layer.
11) In OSI N/W Architecture, the routing is performed by _____
Answer : Network b) Session c) Application d) DataLink
ANSWER : Answer : Network Layer.
Networking

What is the difference between URL instance and URLConnection instance?
ANSWER : A URL instance represents the location of a resource, and a URLConnection instance represents a link for accessing or communicating with the resource at the location.
2) How do I make a connection to URL?
ANSWER : You obtain a URL instance and then invoke openConnection on it.
URLConnection is an abstract class, which means you can't directly create instances of it using a constructor. We have to invoke openConnection method on a URL instance, to get the right kind of connection for your URL.
Eg.

```
URL url;
URLConnection connection;
try{ url = new URL("...");
conection = url.openConnection();
}catch (MalFormedURLException e) { }
```

3) What Is a Socket?
A socket is one end-point of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent.Socket classes are used to represent the connection between a client program and a server program. The java.net package provides two classes--Socket and ServerSocket--which implement the client side of the connection and the server side of the connection, respectively.
What information is needed to create a TCP Socket?
ANSWER : The Local System's IP Address and Port Number.
And the Remote System's IPAddress and Port Number.
5) What are the two important TCP Socket classes?
ANSWER : Socket and ServerSocket.
ServerSocket is used for normal two-way socket communication. Socket class allows us to read and write through the sockets.
getInputStream() and getOutputStream() are the two methods available in Socket class.
When MalformedURLException and UnknownHostException throws?
ANSWER : When the specified URL is not connected then the URL throw MalformedURLException and If InetAddress' methods getByName and getLocalHost are unabletoresolve the host name they throwan UnknownHostException.
Servlets

1) What is the servlet?
ANSWER : Servlets are modules that extend request/response-oriented servers, such as Java-enabled web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database.
Servlets are to servers what applets are to browsers. Unlike applets, however, servlets have no graphical user interface.
2) Whats the advantages using servlets than using CGI?
ANSWER : Servlets provide a way to generate dynamic documents that is both easier to write and faster to run. Servlets also address the problem of doing server-side programming with platform-specific APIs: they are developed with the Java Servlet API, a standard Java extension.
3) What are the uses of Servlets?
ANSWER : A servlet can handle multiple requests concurrently, and can synchronize requests. This allows servlets to support systems such as on-line conferencing.
Servlets can forward requests to other servers and servlets.Thus servlets can be used to balance load among several servers that mirror the same content, and to partition a single logical service over several servers, according to task type or organizational boundaries.
4) Which pakage provides interfaces and classes for writing servlets?
ANSWER : javax
5) Whats the Servlet Interfcae?
ANSWER : The central abstraction in the Servlet API is the Servlet interface. All servlets implement this interface, either directly or, more commonly, by extending a class that implements it such as HttpServlet.
Servlets-->Generic Servlet-->HttpServlet-->MyServlet.
The Servlet interface declares, but does not implement, methods that manage the servlet and its communications with clients. Servlet writers provide some or all of these methods when developing a servlet.
6) When a servlet accepts a call from a client, it receives two objects- What are they?
ANSWER : ServeltRequest: Which encapsulates the communication from the client to the server.
ServletResponse: Whcih encapsulates the communication from the servlet back to the client.
ServletRequest and ServletResponse are interfaces defined by the javax.servlet package.
7) What information that the ServletRequest interface allows the servlet access to?
ANSWER : Information such as the names of the parameters passed in by the client, the protocol (scheme) being used by the client, and the names of the remote host that made the request and the server that received it.

The input stream, ServletInputStream.Servlets use the input stream to get data from clients that use application protocols such as the HTTP POST and PUT methods.

8) What information that the ServletResponse interface gives the servlet methods for replying to the client?

ANSWER : It Allows the servlet to set the content length and MIME type of the reply.
Provides an output stream, ServletOutputStream and a Writer through which the servlet can send the reply data.

9) What is the servlet Lifecycle?

ANSWER : Each servlet has the same life cycle:
A server loads and initializes the servlet (init())
The servlet handles zero or more client requests (service())
The server removes the servlet (destroy())
(some servers do this step only when they shut down)

10) How HTTP Servlet handles client requests?

ANSWER : An HTTP Servlet handles client requests through its service method. The service method supports standard HTTP client requests by dispatching each request to a method designed to handle that request. 1

Encapsulation :
Encapsulation is the mechanism that binds together code and the data it manipulates and keeps both safe from outside interference and misuse.

Inheritance:
Inheritance is the process by which one object acquires the properties of another object.

Polymorphism:
Polymorphism is a feature that allows one interface to be used for a general class of actions. The specific action is determined by the exact nature of actions.

Code Blocks:
Two or more statements which is allowed to be grouped into blocks of code is otherwise called as Code Blocks.This is done by enclosing the statements between opening and closing curly braces.

Floating-point numbers:
Floating-point numbers which is also known as real numbers, are used when evaluating expressions that require fractional precision.

Unicode:
Unicode defines a fully international character set that can represent all of the characters found in all human languages. It is a unification of dozens of character sets, such as Latin, Greek, Arabic and many more.

Booleans:
Java has a simple type called boolean, for logical values. It can have only on of two possible values, true or false.

Casting:
A cast is simply an explicit type conversion. To create a conversion between two incompatible types, you must use a cast.

Arrays:
An array is a group of like-typed variables that are referred to by a common name. Arrays offer a convenient means of grouping related information. Arrays of any type can be created and may have one or more dimension.

Relational Operators:
The relational operators determine the relationship that one operand has to the other. They determine the equality and ordering.

11.*Short-Circuit Logical Operators:*
The secondary versions of the Boolean AND and OR operators are known as short-circuit logical operators. It is represented by || and &&..

12. *Switch:*
The switch statement is Java's multiway branch statement. It provides an easy way to dispatch execution to different parts of your code based on the value of an

experession.

13. *Jump Statements:*

Jump statements are the statements which transfer control to another part of your program. Java Supports three jump statements: break, continue, and return.

14. *Instance Variables:*

The data, or variable, defined within a class are called instance variable.