CONCEPT OF OBJECTS & CLASSES IN JAVA

1)Object- Unique entity which contains data & methods.
2)Real World Object- which really exists.
3)State/Char'- the physical properties of an object.
4)Behaviour- the actions an object is capable of.
5)Method- Function/Behaviour.
6)Abstraction- Hiding unnecessary and displaying necessary details.
7)Encapsulation- wrapping up of variables and methods in a single unit,
8)Methods- Function/Behaviour.
9)Message Passing- Interacting with objects.
10)Class- Description for similar types of objects.
11)Object Factory- Class.
12)Instance- Object.
13)Platform- Hardware and sofware.
14)Source Code- Program written by the user.
15)Byte Code- Code in binary which can run anywhere.
16)Compiler- Which translates the whole program in one go.
17)Interpreter- Which translates a program line by line.
18)Native Executable Code- Platform dependent progs.
19)JVM- Java virtual Machine, collection of java's own programs required for execution.
20)API- Application Programming Interface. Java's own classes and functions we can use in our program.
21)IDE- Integrated Development Environment – where we can type, edit, save, run and organize java programs.
22)Class "Object"- Super class of all the objects in Java.
23)Standalone application- Which run on a single comp.
24)Applet- Which runs on a web browser.
25)Initial Class- Which contains main().
26)Object Code- Got after compilation.
27)Machine Code- Got after compilation.
28)Jar File- Collection of Java programs.
29)Documentation- Details about the program.
30)Polymorphism- Different behaviour under different situations.
31)Inheritance- When a class acquired properties of another class.
32)Recursion- when a function calls itself.
33)Interface- Template/Design of a class. only with variable and function names.
34)Single Line Comment- //remark or note.
35)Multi-line Comments- /* many lines */
36)Documentation Comments- /** remarks */ just before the function header. The are displayed in the method call window in blueJ.
37) Glossary- List of variables, their types and uses.
38)Abstraction using Classes- private.
39)Encapsulation using Classes- class { ... }
40)Polymorphism using Classes- Function overloading
41)Inheritance using Classes- extends.
42)WORA (& some other features)- Write once Run anywhere, light weight, advanced features, open product.

FUNCTIONS & CONSTRUCTORS

1. Named unit of a program. Method.
2. Reusability, Modularity
3. Using functions to solve a problem.
4. A function can exist only inside a class.
5. Time taken to go to and return from a function.
6. Overheads.
7. Prototype- First line
8. Signature- formal arguments
9. Access Specifier- Where all (or not) can be accessed outside a class.
10. Modifier- Can be accessed with or without objects.
11. Return Type- The data type of the value a function returns
12. Formal argument- variables present in () after the function name.
13. Function declaration- Prototype.
14. Function header- Prototype.
15. Actual arguments- variables present in () in the function call.
16. Extended Name- Function name+signature.
17. Jumps back & ignores rest of the code – Can be more than 1- can only carry 1 value
18. Must have something on the lhs.
19. Instance method- non static, Class method - Static?
20. Advantage- can be called without objects, disadvantage- can access static members only.
21. Show the difference between call by value and reference using an example?
22. Why are the changes reflected in pass by reference and why not in pass by value?
23. Only of mutable classes.
24. On running the program (e.g. java add 5 10)
25. computational-manupulative-procedural. (POP)
26. Pure don't and impure change the state(instance variables) of an object.
27. Accessor, getter.
28. Mutator, setter.
29. Having 2 or more function with the same name but diff signature.
30. Number, Type and order of arguments.
31. Argument name and return type.
32. The actual arguments.
33. Same name can be used for many functions with the same purpose.
34. Which contains main().
35. double simpleInterest(double p, double r, double t);
36. double force(double mass, double acc), and double time(double d, double s)
37. char charAt(int index), int indexOf(char/string)
38. double Sqrt(double num), int max(int n1, int n2) OR double max(double n1, double n2)
39. boolean isPrime(int num)
40. boolean isPalindrome(String s)
41. double force(double mass, double acc)
42. void anyName()
43. void anyName(int n)

44. double area(double side), double area(double length, double breadth), double area(double a, double b, double c).
45. double add(double a, double b), double add(double a, int b), double add(int a, double b), int add(int a, int b).
46. An error we get after defining overloaded functions with the same signature.

CONSTRUCTORS

47. Member function with the same name as that of a class. Initialization.
48. Object creation – class-name object-name=new constructor(values)
49. Object of the class cannot be accessed outside the class.
50. A non-parameterized constructor. In the absence of any constructor.
51. false.
52. null (ascii code 0)
53. null (null literal)
54. Constructor- same name, not return type, should be public, called automatically.
55. No (yes in newer versions of java, but write no only)
56. Having more than one constructor. An object can be created with or without giving values.
57. Default, parameterized, (and copy). Egs in notes.
58. In the absence of any constructor.
59. To identify the instance variable when a local variable has the same name. *Give example from notes.*
60. An object created but not stored. E.g.- new A().
61. Written in notes. Select 6 and write.
62. A function called when an object is no longer required and removed from the memory by the compiler. finalize().
63. Cleaning up of the memory.
64. by writing *this().*
65. constructors – any number as long as the arguments are different, destructors- only 1.
66. variables – constructor – input() – processing() – output().
67. Which accepts an object of its own type.

UNDERSTANDING LIBRARY CLASSES

68. Inbuilt classes provided with Java.
69. io –util – lang - net – applet –swing – awt.
70. System=class, out=object, println()=function.
71. print()-keeps the cursor on the same line after printing. println()- takes to the next line after printing.
72. Accepts ascii values only.
73. Flow of bytes. (incoming or outgoing)
74. Input stream and output stream.
75. System.out, System.in, System.err.
76. Character oriented and byte oriented.
77. Character oriented – keyboard, byte oriented- Scanner.
78. See notes.
79. Syntax – logical and runtime.
80. A runtime error.
81. A wrong value
82. Managing runtime errors.
83. Program doesn't crash.
84. Perform exception handling.
85. An excpetion.
86. On th event of an exception,
87. On executing the "throw" statement.
88. (1) always executed (2) Should be the last.
89. "throw"- explicitly raise an exception, "throws" – avoid try-catch in the program.
90. ArrayIndexOutOfBoundException, StringIndexOutOfBoundException,
91. Yes
92. By using different exception names.
93. A class equivalent of a primitive data type.
94. Functionality.
95. Same as the primitive type but starts with a capital letter. (int=Integer, char=Character)
96. parseInt()- convert string to integer, toString()- convert to string.
97. String- processing immutable strings, StringBuffer- processing mutable strings, StringTokenizer- word by word processing of a sentence.
98. append(), insert(), delete(), reverse(), capacity().
99. StringBuffer- mutable, call by reference, different functions.
100. Successively calling 2 or more functions. Eg. Sop(s.concat("S").length());
101. Collection of classes
102. Saves memory- all classes need not be loaded in the memory for all the programs.
103. package package-name; at the top of the program.
104. import package-name; at the top of the program.
105. All the classes of the package.
106. Only that class can be used in the program.
107. sqrt(), pow(), isLetter()..., we don't need to create objects all the time.
108. Sop(new Date());
109. extends.
110. Reusability.
111. the main class and the one made from it. base and derived.
112. Which cannot be instantiated. (Objects cannot be made)
113. Template of a class.
114. They are final.
115. the are abstract (cannot be defined)
116. Easier way to process input. next(), nextLine(), nextInt()...
117. pw.println("name") (after PrintWriter pw=new PrintWriter(System.out, true))
118. display immediately.
119. Scanner- util, printwriter- io.
120. (a) 3 parts of an object creation statement – declaration – instantiation – initialization. E.g. A obj=new A(5,10) the declaration=A obj, instantiation=new A(), initialization=A(5,10). (b) Null reference= object declared but not instantiated.

END OF THEORY ANSWERS