(Object Oriented Programming with Java)

OOP Basics

1. Define **Object Oriented Programming (OOP)**. or What does Object Oriented Programming (OOP) mean? [2011] [2013] [2013 C++] [2014] [2015]
2. What are the differences between **OOP** and **Procedural Programming**? [2011]
3. Why **OOP** is effective over **structured programming**? Explain. Hw 3 [2014 C++]
4. Briefly discuss the **properties** of OOP language with example. [2014 C++]
5. What are **features** of OOP? Describe them. Or Can you write the **basic features** of OOP? [2011] [2013] [2013 C++]
6. Write some features of OOP. [2016 C++]
7. Distinguish between **encapsulation** and **polymorphism**. [2013]
8. Explain the following concepts of Object Oriented Programming: [2014]
   i. **Data abstraction**
   ii. **Polymorphism**
9. How is **polymorphism** achieved at (a) compile time and (b) run time? Hw 3 [2015] [2012 C++]
10. Distinguish between them. [2012 C++]
11. Explain different types of polymorphism in Java. [2013 C++]
12. Distinguish between static binding and dynamic binding. [2013 C++]
13. Write the **applications** of OOP. [2014]
14. How are **data and functions organized** in an object-oriented program? [2015]
15. How are **data and functions organized** in OOP? Explain. Hw 3 [2013 C++]
16. Differentiate between **Java** and **C++** as object oriented programming language. [2013]

Data Types & Operators

17. What does meant by **literals**? Shortly explain **integer**, **floating-point**, **boolean**, **character** and **string literals** used in Java programming. Hw 3 [2014]
18. What are the different **data types** used in Java? Give examples. Hw 3 [2014]
19. Discuss the **scope** and **lifetime** of a **variable**. Hw 3 [2014]
20. Write a fragment of code that make use of the **shorthand operators** like += and -= Hw 3 [2014 C++]
21. What is the difference between comparing with **== operator** and **equals()** method? [2017]
22. How can you create **prefix** and **postfix** forms of the increment and decrement **operators**? Hw 3 [2015]
23. How does **binary operator** operate? Explain with example. Hw 3 [2015]

Basics, Class & Methods

24. In **System.out.println()** what is System, out and println? Explain. Hw 3 [2013]
25. Give and explain the basic **structure** of **Java program**. [2013]
26. Explain the **steps** to be followed for **executing** a **Java program**. [2014]
27. What is **byte-code**? [2014]
28. Show and discuss the **general form** of a **java class**. Hw 3 [2014]
29. What does it mean by "**String is Immutable in Java**"? Why String is **Immutable** in Java? [2017]
30. How is Java **more secured** than other languages? (HW 2) [2011]
31. Why is Java known as **platform-independent language**? (HW 2) [2011]
32. What are **Java package** and **Java applets**? (HW 2) [2011]
33. What type of **naming convention** should be followed when declaring package, variables, methods and constant? [Hw 2] [2011]
34. In what ways do you initialize of **instance** fields? [2014]
35. What is a **class**? [2014] [2013] [2012 C++] [2011]
36. Discuss class and object with example. [2013 C++]
37. What is **reference**? Describe with example. [2013 C++]
38. What are the differences between '**class**' and '**structure**'? [2011]
39. Differentiate between **structure** and **class** with an example. [2016 C++]
40. How class accomplishes **data hiding**? [2014]
41. How **data hiding** is accomplished in Java? Explain. Hw 3 [2013 C++]
42. What is **inner class**? Give example. [2014]
43. What are **wrapper classes**? Why should we need a wrapper class? [2014]
44. How are **memory allocated** for different type of **members of a class**? [2011]
45. Give an example of '**nested class member**'. [2011]
46. What are **objects**? or What is an object? [2014] [2013] [2012 C++] [2011]
47. How are objects **created** from a class? [2014]
48. What kind of things **can become an object** in OOP? Hw 3 [2015]
49. How can you create an **initialized array** of objects? Hw 3 [2015] [2013 C++]
50. Is it possible to have arrays of objects? If possible, how? [2016 C++]
51. What are the **characteristics** of an **object**? [2013]
52. Explain about "**object down casting**" with example. [2013]
53. Differentiate among **instance variable**, **class** variable and **local** variable. [2013]
54. What are the differences between **static** and **non-static methods**? [2013]
55. What is **access specifier**? [2013]
56. Is there any way to **access private member** of a class without taking help of own member function of that class? Explain your answer with an example. [2011]
57. Discuss the implications of deriving a class from an existing class by the 'public' and 'protected' access specifiers with examples. [2016 C++]

58. What is **operator overloading**? Why is it **necessary**? [2011]

59. What do you mean by operator overloading? What are the restrictions applied for operator overloading? [2012 C++]

60. Why **overloading** sometimes causes ambiguity? Describe with example. [2011] [2016 C++]

61. Which type of **ambiguity** may arise? [2011]

62. What are '**unary**' and '**binary**' **operator overloading**? [2011]

63. Give an **example** of '**unary**' operator overloading. [2011]

64. Differentiate between **overriding** and **overloading**. [2014]

65. What is **method overloading** and **method overriding**? Give an example. [2017]

66. What is **new** and **delete**? What are their advantages? [2013 C++]

Constructor

67. What is a **constructor**? What are its special properties? [2014]

68. How do we invoke a constructor? Explain with example. [2014]

69. What are the differences between a constructor and a method? [2014]

70. What is **constructor overloading**? What are the reasons to overload a constructor? [2016 C++]

Inheritance

71. What is meant by **inheritence.** Discuss with example. [2013] [2013 C++] [2014]

72. What are its **advantages**? [2014]

73. Explain the **various form** of inheritance. Give an example of each. [2014]

74. Explain different **types/forms** of inheritence with block diagram and examples. Hw 3 [2016]

75. Write a program to implement **multiple inheritance**. [2013]

76. What is the **ambiguity** that arises in **multiple inheritence**? How it can be overcome? Explain with example. Hw 3 [2016 C++]

77. Describe a scenario in which **multi-level inheritance** can cause **ambiguity**. And how this ambiguity can be solved. Hw 3 [2014 C++]

78. What are **multiple**, **multilevel** and **hybrid inheritance**? [2011]

79. What is multiple inheritance (virtual inheritance)? What are its advantages and disadvantages? [2012 C++]

80. How are **this()** and **super()** used with **constructors**? [2013]

81. What is "**this**"? Explain with an example. [2013 C++]

82. Explain the **use** of **super**() with example. [2014]

83. When do you declare a method or class **final**? [2014]

84. Explain why do need to use **'final'** with inheritance. [2014]

85. What is **final variable** and **final method**? Write-down the reasons to use these. [2013]

86. Explain how '**virtual inheritance**' can solve the problem that is caused when any member of base class may be inherited in different ways to a 'high level derived class'. [2011]

Interface

87. Define **abstract class**, **concrete class** and **interface**. [2014] [2017]
88. What are the **differences** between abstract class and interface? [2017] [2013]
89. Explain the **use** of **abstract class**. [2014] [2017]
90. Give the **syntax of interface**. Differentiate between **overriding** and **overloading**. [2014]
91. What are the rules of defining a **functional interface**? [2017]

Exception

92. What is an **exception**? [2013 C++]
93. What is **exception handling**? [2014 C++] [2015 C++]
94. Why do we need to handle exception? [2013 C++]
95. Write the general form of exception handling. [2015 C++]
96. Discuss the **basic structure** of exception handling in Java. [2013]
97. List five common **examples** of exceptions. [2013]
98. Can you explain the use of **finally** block in java exception handling mechanism. [2013]
99. State the tasks of **throw** and **throws**. [2013]
100. How can you **rethrow** an **exception**? [2014 C++]
101. Explain how exception is handled by using '**try**', '**catch**' and '**trhow**' ? (or Briefly discuss) [2011] [2014 C++]
102. What is the mechanism to catch all the exceptions? [2014 C++]
103. Explain the following **Java Keywords** with appropriate example. [2017]
    a. try
    b. catch
    c. throw
    d. throws
    e. finally
104. How can you create your **own type of exception**? [2017]

Event

105. Describe the operation of an **event delegation model**. [2013]
106. With suitable block diagram, explain **Delegation Event Model** in java. [2017]
107. Explain the following Java terms: [2013]
    a. Event
    b. Listener

c. **Adapter**

GUI

108. Distinguish between **Abstract Window Toolkit (AWT)** components and **Java Swing** components. [2014]

109. Distinguish between **Java swing** and **awt classes**. [2013]

110. Why is **swing** preferable for GUI programming **over AWT**? [2017]

111. Explain why support for **concurrency** in necessary for any programming language that is used to build a **Graphical User Interface (GUI)**. [2013]

112. Write a simple **JavaFX skeleton application** and discuss its key **components**. [2017]

File

113. Write a java program to display a **file image** on web-browser. [2014]

114. Differentiate between **text file** and **binary file**. [2013]

115. Where should you write a **file closing statement** to close a **file stream**. How can you **open** a file stream so that it will **close automatically** when it is no longer in use? [2017]

116. What are the differences between **Byte Streams** and **Character Streams**. [2017]

117. How can you **read** from and **write** to a **text file**? Discuss. [2015 C++]

118. Write a short program that will use Java I/O library to **write** n random numbers **to a file**. Hw 3 [2014 C++]

119. What is the difference between **opening a file** with constructor function and opening a file with 'open()' function? [2016 C++]

120. Shortly discuss **random access** from a **file** with example. [2015 C++]

121. What is **stream**? [2013 C++]

122. Name the streams generally used file I/O. [2013 C++]

Networking

123. Discuss the process of creation of **server** and **client sockets** with exceptions handled explicitly with a suitable example. [2017]

124. How do you **connect** to a **URL resource** in Java programming language? [2017]

Others

125. Explain the accessing mechanism of data members and member functions in case of (i) inside main () function and (ii) inside a member function(or method!?) of the same class. [2013 C++]

126. What is the difference between **start()** and **run()** methods of **Thread class**? [2017]

127. What happens when an **Exception** occurs in a **thread**? [2017]

128. How do you **share data** between **two threads in Java**? [2017]

129. Define **multithreaded programming**. [2014]

130. What do you mean by **multithreading**? [2013]

131. What is '**function template**'? [2011]

132. Give an **example** of 'function template'. [2011]

133. Why is **main** method static in Java? Explain. Hw 3 [2013]

134. When do you declare a member of **a class static**? [2013]

135. How is **main()** method declared in java. Discuss briefly the meaning of each part of the main() method declaration. [2013]

136. What are the difference between the constants **7**, **'7'** and **"7"** ? [2013]

137. Write a java program to print the **average values of five array**s of different lengths, each of which has hold different number of integer values respectively, using the **method average()** in your program. [2013]

138. How can you create an **ArrayList**? How do you add remove elements from it? How can you obtain an Array from an ArrayList? [2017]

139. Can you pass **List<String>** to a method which accepts **List<Object>**. [2017]

140. How to write a **generic method** which accepts **generic argument** and return **Generic Type**? [2017]

141. What is **applet**? Explain the **lifecycle** of an applet. [2013]

142. Describe the differences between a **Java applet** and a **Java application**. [2013]

143. What are the differences between **Java Applet Web Application** and **JavaScript**? [2014]

144. Define **Java Bean**? What are the advantages of **Java Beans**? [2014]

145. What are advantages of using '**static member variable**' and '**static member function**' of a class? [2011]

146. What is meant by **data binding**? Hw 3 [2015]

147. What is the difference between **early binding** and **late binding**? [2011] [2016 C++]

148. Define them with example. [2012 C++]

149. What is the **default argument**? [2011] [2016 C++]

150. What are the advantages of default arguments? Give example. [2016 C++]