**Name :**

**Employee ID :**

**Date :**

Weightage:

MCQs: 10% Fill in the blanks: 20%

Code based questions: 30% Subjective questions: 40%

**MCQs**

*NOTE:*

1. *Each MCQ could have one or more correct answers. Partial correct answers will not be considered.*
2. *Write down your chosen options at the bottom of each MCQ where a provision is given for “Answer:”.*
3. *Think and make your choices before writing the answer. Repeated overwriting/correction of answers must be avoided.*
4. **Which of the following are the features of the Eclipse Refactoring Tool?**
5. Hardcoded literals (values) can be converted to class level constants.
6. Change the name of a variable or method by taking care of their usage in other parts of the application code.
7. Generate boiler-plate code for constructors and certain types of methods.
8. Format the code for proper indentation

*Answer:*

1. **Which of the following statements are true?**
2. A protected method can be overridden in the child class with public access.
3. A private method can be overridden in the child class.
4. A method with default access can be overridden in the child class, which is in a package that is different from its parent class.
5. A public method can be overridden in the child class with public access.

*Answer:*

1. **Which of the following statements are true?**
2. An unchecked exception must either directly or indirectly extend RuntimeException class.
3. An unchecked exception **cannot** be included in the throws clause of a method.
4. Listing or not listing an unchecked exception in a throws clause makes no difference.
5. Unlike a checked exception a catch block cannot be used to handle an unchecked exception.
6. A *finally* block can have *try-catch* blocks nested in it.

*Answer:*

1. **Which of the following statements are true when an object of a POJO is stored in an *ArrayList* ?**
2. The implementation of *equals()* method of the POJO has ***no*** impact on the functionality of the method *indexOf(Object o)*
3. The implementation of *equals()* method of the POJO has an impact on the functionality of the method *removeAll(Collection c)*
4. The implementation of *equals()* method of the POJO has an impact on the functionality of the method *contains(Object o)*
5. The implementation of *hashCode()* method of the POJO has an impact on the functionality of the method *add((E e)*
6. The implementation of *hashCode()* method of the POJO has an impact on the functionality of the method *retainAll(Collection c)*

*Answer:*

**Fill in the blanks:**

*NOTE:*

1. *Think and finalize your answer before filling in the blanks.*
2. *Avoid overwriting/corrections*
3. *Ensure your writing is legible. (legible = clear enough to read)*
4. For an attribute of a class to be accessible only to sub-classes in the same package \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ access level is used.
5. StringBuilder provides better performance over StringBuffer because \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
6. The method of Boolean class to be used to convert a string value *“true”* into a Boolean object is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
7. To prevent data from being shared during the process of serialization the attribute of the serializable object should be declared as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

**Code based Questions:**

*NOTE:**Write your answer where a provision is given as “Answer:”.*

1. What is the output of the below code?

Integer[][] arr = new Integer[3][4];

System.out.print(arr[2][1];

*Answer:*

1. What is the output of the below code?

public class Arraylist {

public static void main(String args[]) {

ArrayList obj = new ArrayList();

obj.add("J");

obj.add("D");

obj.add("C");

obj.add(1, "B");

System.out.println(obj);

}

}

*Answer:*

1. What will be the output of below code?

Object charSet = new String("instanceof");

List<String> strList = new ArrayList<>();

Object intObj = Integer.valueOf(25);

Set<Integer> intSet = new HashSet<>();

System.out.println( (charSet **instanceof** Collection)

+ ", "

+ (strList **instanceof** Collection)

+ ", "

+ (intObj **instanceof** Collection)

+ ", "

+ (intSet **instanceof** Collection));

*Answer:*

1. What is the output of below code when it is executed without any command line arguments?

public class ExceptionHandling ***{***

public static void main(String**...**ZettaMine) **{**

Object[] objArray = { 1, "Five", Integer.valueOf("9") };

try {

int val = Integer.valueOf(objArray[1].toString());

System.out.println(val);

} catch (Exception e) {

System.out.println(ZettaMine.length);

}

**}** *//end of main*

***}*** *//end of class*

*Answer:*

**Subjective Questions**

*NOTE:*

1. *Start answering below these questions.*
2. *Use additional sheets as required.*
3. *Number the additional sheets appropriately.*
4. *Write your name and employee id on each additional sheet used.*
5. Explain the purpose of serialization.
6. Explain the difference between the interfaces *Comparable* and *Comparator*.
7. How is a plain try block different from try-with-resources?

.