411 MID TERM SAMPLE FA ’22 (**solutions**)

1. Which of these is an incorrect array declaration?

**a) intarr[]=newint[5]** b) int [] arr = new int[5]

c) int arr[] = new int[5] **d) int arr[] = int [5] new**

2. What will this code print?

int arr[] = new int [5];

System.out.print(arr[0]);

**a) 0** b) value stored in arr[0] c) 00000 d) Class name@hashcode in hexadecimalform

3. What is the output of this program?

  class array\_output {

      public static void main(String args[])    {

          int array\_variable [] = new int[10];

   for (int i = 0; i < 10; ++i) {

              array\_variable[i] = i;

              System.out.print(array\_variable[i] + " ");

              i++;

          }

      }

  }

a) **0 2 4 6 8** b) 1 3 5 7 9 c) 0 1 2 3 4 5 6 7 8 9 d) 1 2 3 4 5 6 7 8 9 10

4. Given a binary search routine, if lower is the first subscript and upper is the last subscript, then the array item in the middle of that array portion is at subscript

**a) (lower + upper)/2** b) (upper — lower)/2 c) (lower — upper)/2 d) lower + upper/2

5. Which is the valid declaration within an interface definition?

**a) public double methoda();** b) public final double methoda();

c) static void methoda(double d1); d) protected void methoda(double d1);

6. True or **False**? A private base method can be overridden by some derived public method.

7. Which of these keywords is not a part of exception handling?

a) try b) finally **c) thrown** d) catch

8. The process of converting one date type to another is called\_\_\_\_\_\_\_\_\_\_.

1. Translating    **b) Casting**c) Compiling    d) Declaring

9. Which of the following is the correct declaration statement in java program?

a) int num=int[5] b) int num=new num[5] **c) int[] num=new int[5]** d) None

10. What line of code should replace the missing statement to make this program compile?

/\* Missing Statement ? \*/  
public class foo {  
   public static void main(String[]args)throws Exceptionn {  
       java.io.PrintWriter out = new java.io.PrintWriter();   
       new java.io.OutputStreamWriter(System.out);   
       out.println("Hello");   
   }   
 }

* 1. **No statement required.** b) import java.io.\*; c) include java.io.\*; d) import java.io.PrintWriter;

11. \_\_\_\_\_\_\_\_\_\_\_\_method is used to find the nth no. of character of given string s1.

1. 1.index(n) b) s1.substring(n) c) s1.length()  **d) s1.charAt(n)**

12. Which of these can be overloaded?

a) Methods b) Constructors **c) both a and b** d) None can be overloaded

13. The concept of multiple inheritance is implemented in Java by?

I.   Extending two or more classes.

II.  Extending one class and implementing one or more interfaces.

III. Implementing two or more interfaces.

a) Only (II) b) (I) and (II) **c) (II) and (III)** d) Only (I) e) Only (III)

14. True or **false**? Private members of a class are inherited in the sub class.

15. **True** or false? A method of a super class with a default access modifier can be overridden as protected or public but not as private.

16. What will be the output of the following program?

class X {

void method(int a) {

       System.out.println("ONE");

 }

 void method(double d)  {

       System.out.println("TWO");

 }

}

class Y extends X {

  @Override

 void method(double d) {

       System.out.println("THREE");

  }

}

public class MainClass {

   public static void main(String[] args)  {

       new Y().method(100);

   }

}

1. **ONE**  b) TWO c) THREE d) ONE TWO e) Error

17. If a superclass does not have a default constructor,

**a) then a class that inherits from it, must call one of the constructors that the superclass does have**

b) then a class that inherits from it, must contain the default constructor for the superclass

c) then a class that inherits from it, does not inherit the data member fields from the superclass

d) then a class that inherits from it, must initialize the superclass values

18. The binary search algorithm

**a) will cut the portion of the array being searched in half each time the loop fails to locate the search value**

b) will have an average of N/2 comparisons, where N is the number of elements in the array

c) is less efficient than the sequential search algorithm

19. All abstract classes must contain abstract methods. a. True **b. False**

Coding and Definitions:

20. What are some basic differences between an Array and ArrayList?

**The array represents a basic functionality whereas the ArrayList is part of collection framework in Java. Therefore array members are accessed using [ ], whilst an ArrayList has a set of methods to access elements and have the ability to modify them. Arraylists are also dynamic in memory and can grow and shrink at runtime.**

21. What is the major difference between an abstract class and an interface?

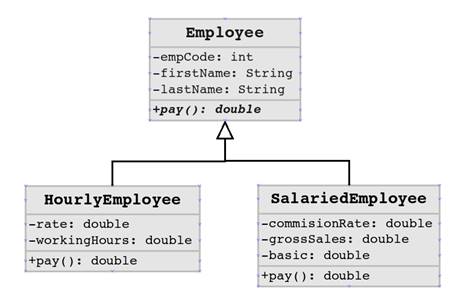
**Abstract class is more *expensive* to use, as there is a look-up to do when you inherit from them. Can define a behavior for common classes.**

**An interface is a contract, a generic pattern of methods with just signatures allowing any class implement its methods in a way that it wishes to do.**

22. Define Polymorphism? What are some of its abilities/usefulness?

**Means a state of having many shapes or the capacity to take on various forms. Ability to describe and process objects of various types and classes through a single, uniform interface (base object).**

**Sample diagram of sub types follows.**



23. Name a minimum of 4 data types (2 primitive and 2 reference types) supported in java?

**int, double, Integer, Double**

24. Create and define a method called **popList** that gets passed an **arraylist** called list that will accept any numeric value type passed in. The method definition should assign 5 integers to the arraylist via a loop storing the values of 1,2,3,4 and 5 respectively.

static void popList (ArrayList<Number> list) {

for( int i=1; i<=list.size(); ++i)

list.add(i);

}

25. Code the sort method below to sort an arraylist in ascending order? (DO NOT USE ANY PRE DEFINED METHODS)

public class aListClass{

  public static void main(String[] args) {

        ArrayList<Integer> values = new ArrayList<>();

        values.add(1);

        values.add(41);

        values.add(3);

        sort(values);

        for (int i = 0; i < values.size(); i++)

            System.out.println(values.get(i));

    }

    public static void sort(ArrayList<Integer> aList) {

    int temp;

        for (int i = 0; i < aList.size() - 1; i++) {

            for (int j = i + 1; j < aList.size(); j++) {

                if (aList.get(i) > (aList.get(j))) {

                    temp = aList.get(i);

                    aList.set(i, aList.get(j)); //swap

                    aList.set(j, temp);         //values

                }

            }

        }

}

}