Unit 3

Q. No.	Question/Answer	Marks
1.	What is the value returned by function compareTo(), if the invoking strong is less than the string compared?	1
Ans:	compareTo() function returns a value less than zero if the invoking string is less than the other string being compared.	
2.	In which memory a String is stored, when we create a string using new operator?	1
Ans	Heap Memory	
3.	Which of the classes are the direct subclasses of the Throwable class?	1
Ans:	Error and Exception class.	
4.	What does toString method do?	1
Ans:	toString method that returns a string representing the value of the object.	
5.	Name two built-in methods in Arrays class.	2
Ans:	Arrays.toString, Arrays.fill().	
6.	What is utility class? And name one utility class.	2
Ans:	All methods are static methods. Array/Math is one of the utility class.	
7.	What is the use of using super()?	2
Ans:	When Java makes an implicit call to the superclass constructor, it calls the default constructor. If we want to invoke a superclass constructor that takes arguments, we have to do so explicitly using super( ).	
8.	Say true or false: i) In Java, array is a reference type. ii) In Java, arrays cannot be assigned. iii) Array elements are always on the heap. iv) A class can extend only one other class	4
Ans:	i) True ii) false iii) true iv) true	
9.	List two differences between Method Overloading and Overriding.	4

Ans:	*Overloading:* Methods with the same name but a different number or type of arguments. Performed within the class. *Overriding:* Methods with the same name but with same number or type of arguments. Occurs in two classes that have is-a relationship.	

10.	State the errors in the following program.
	
	Class Test{
	      private final int a = 10;{
	            a = 20
	      }
	      public test(int x){
	            System.out.println("ok");
	      }
	      public int sqr(){
	            System.out.println(a * a);
	      }
	      public int foo(double x, double y){
	            return x + y;
	      }
	      public void bar(){
	            System.out.println(a.this);
	      }
	}
		6

Ans:	Class Test{
	      private final int a = 10;{
	            a = 20; **// Compile time Error; cannot assign to a final field.**
	      }
	    // Is not constructor or return type missing
	      public test(int x){
	            System.out.println("ok");
	      }
	      public int sqr(){
	            system.out.println(a * a);
	       // return statement missing
	      }	

```
        public int foo(double x, double y){
                return x + y;
            // cannot convert double to int
        }
        public void bar(){
                System.out.println(a.this);
            // should be this.a and not a.this
        }
}
```