Unit 3

Q. No.	**Question/Answer**	Marks
1.	What is interface?	1
Ans:	An interface is a reference type, similar to a class that can contain only constants, method signatures, default methods, static methods, and nested types.	
2.	What is the use of the clone() method?	1
Ans	The clone() method is used to create a copy from an existing object.	
3.	_____ class is the top of the class hierarchy tree.	1
Ans:	Object class.	

4.	What is wrong with the below interface? And why?
	public interface Sample{
	void print(){
	System.out.println("In Print method");
	}
	}
		2

Ans:	It has a method implementation in it. In interface only default and static methods have implementations.	

5.	Is the below interface a valid or not? And why?
	public interface Example
	{
	}
		2

Ans:	Yes. Methods are not required. Empty interfaces can be used as types and to mark classes without requiring any particular method implementations.	
6.	What is the use of interface?	2
Ans:	• When we want to specify the behavior of a particular data type, but not concerned about who implements its behavior. • When we want to take advantage of multiple inheritance of type.	
7.	Say True or False: i) In interface, method bodies exist only for default methods and static methods. ii) abstract class can have constructors and static methods. iii) interface cannot be instantiated.	3

<table>
<tr><td>Ans:</td><td>i) True
ii) True
iii) True</td><td></td></tr>
<tr><td>8.</td><td>Explain overriding toString() method with an example program.</td><td>4</td></tr>
<tr><td>Ans:</td><td>
<pre>
public class Example5
{
	public static void main(String[] args)
	{
		Rect r1=new Rect(20,10);
		System.out.println(r1); //toString
		Object o=r1;
		System.out.println(o);
	}
}
class Rect
{
	double length;
	double breadth;
	public Rect(double l,double b)
	{
		length=l;
		breadth=b;
	}
	public String toString() //override
	{
		return length+" "+breadth;
	}
}
</pre>
</td><td></td></tr>
<tr><td>9.</td><td>Name 4 methods that are inherited from Object class.</td><td>4</td></tr>
<tr><td>Ans:</td><td>equals(),toString(),hashCode() and clone().</td><td></td></tr>
<tr><td>10.</td><td>Difference between interface and abstract class.</td><td>6</td></tr>
<tr><td>Ans:</td><td>
<table>
<tr><th>Sl.No</th><th>Interface</th><th>Abstract class</th></tr>
<tr><td>1</td><td>Supports multiple inheritance.</td><td>Does not support multiple inheritance.</td></tr>
<tr><td>2</td><td>Does not contain data member ,constuctor</td><td>Contain data member ,constuctor</td></tr>
<tr><td>3</td><td>Contains only incomplete (only signature)member.</td><td>Contains only complete and incomplete member.</td></tr>
<tr><td>4</td><td>Cannot have access modifiers by default everything is assumed as public.</td><td>Can Contain access modifiers for the methods, properties.</td></tr>
</table>
</td><td></td></tr>
</table>

	5	Cannot have static member.	Only complete member can be static.	