1- Consider the following class:

```
    public class Project {
        private int projectId;
        private String projectName;
     //Assuming getters & setters are available
       }
```

What will be the output of following driver class:

```
public class ProjectDriver {
    public static void main(String[] args)  {
        Project project=new Project();
        project.setProjectName("FinnOne");
        System.out.println(project.getProjectName());
        changeProjectName(project,"FinnAxia");
        System.out.println(project.getProjectName());
    }
    public static void changeProjectName(Project p,String projectName)
    {
        p.setProjectName(projectName);
    }
}
```

2- Consider the following UserEntity class

```
       class UserEntity{
           private int id;
           private String name;
       //getters setters provided
       }
```

What will be the output of following code snippet:

```
public class TestObjectConcept {
    public static void main(String[] args) {
        UserEntity u1=new UserEntity();
        UserEntity u2=new UserEntity();
        u1.setId(101);
        u1.setName("Bob");
        u2=u1;
        u1.setId(404);
        u2.setName("Joey");
        System.out.println("user1:   "+u1.getId()+"   "+u1.getName());
        System.out.println("user2:    "+u2.getId()+"  "+u2.getName());
    }
}
```

3- What is the difference between method and constructor?

4- What will be the output:

```
class Skill
{
    String skillName;
    String description;
    Skill(String skillName,String description){
        this.description=description;
        this.skillName=skillName;
    }
    public String getSkillName() {
        return skillName;
    }
    public String getDescription() {
        return description;
    }
}
public class Assessment {
    public static void main(String[] args) {
     Skill [] skillArr={
                            new Skill("Java","Core"),
                            new Skill("C++","Basic"),
                            new Skill()
                           };
        System.out.println(skillArr[1].getSkillName());
    }
}
```

5- Complete the code as stated:

```
    class Language{
        String languageName;
        int experience; //in years
        Language(String languageName,int experience)
        {
            this.languageName=languageName;
            this.experience=experience;
        }
    }
    public class TestDemo {
        public static void main(String[] args) {
            Language []languages=new Language[5];
      //add code to initialize the array with 5 different languages
    }
    }
```

6- Consider following class:

```
    public class Project {
        private int projectId;
        private String projectName;
        private String projectLead;
        //Assuming getters & setters are available
}
```

Define following method:

void updateProjectLead(Project[] projects,String leadName)

{

//logic to update projectLead with the leadName of the project if no project-lead is assigned to it

}

7- State usage of this keyword.

8- State usage of static keyword.

9- Explain immutable nature of String with appropriate code segment.

10- What will be the output:

```
public class TestDemo {
    public static void main(String[] args) {
        String s1=new String("Today is beautiful");
        String s2="Today is beautiful";
        String s3=new String("Today is beautiful");
        System.out.println(s1.equals(s2));
        System.out.println(s1.equals(s3));
        System.out.println(s1==s2);
        System.out.println(s2=="Today is beautiful");
    }}
```