Experiment No 6

Write a program in Java to create a player class. Inherit the classes Cricket_player, football_player and Hockey player from player class.

Objectives: To learn the use of inheritance and constructor in java.

Theory:

Inheritance:-

Inheritance is one of the cornerstones of object-oriented programming because it allows the creation of hierarchical classifications. Using inheritance, you can create a general class that defines attributes common to a set of related items. This class can then be inherited by other, more specific classes, each adding those things that are unique to it. In the terminology of Java, a class that is inherited is called a ***superclass***. The class that does the inheriting is called a ***subclass***. Therefore, a subclass is a specialized version of a superclass. It inherits all of the instance variables and methods defined by thesuperclass and adds its own, unique elements.

Inheritance Basics

To inherit a class, you simply incorporate the definition of one class into another by using the **extends** keyword. To see how, let's begin with a short example. The following program creates a superclass called **A** and a subclass called **B**. Notice how the keyword **extends** is used to create a subclass of **A**.

```
// A simple example of inheritance.
// Create a superclass
class A
 {
int i, j;
void showij()
{
System.out.println("i and j: " + i + " " + j);
}
}
```

// Create a subclass by extending class A.

```
class B extends A
{
int k;
void showk()
{
System.out.println("k: " + k);
}
void sum()
{
System.out.println("i+j+k: " (i+j+k));
}
}
```

As you can see, the subclass **B** includes all of the members of its superclass, **A**. This is why **subOb** can access **i** and **j** and call **showij( )**. Also, inside **sum( )**, **i** and **j** can be referred to directly, as if theywere part of **B**.

Even though **A** is a superclass for **B**, it is also a completely independent, stand-alone class. Being asuperclass for a subclass does not mean that the superclass cannot be used by itself. Further, a subclass can be a superclass for another subclass.

The general form of a **class** declaration that inherits a superclass is shown here:

class *subclass-name* extends *superclass-name*

```
{
// body of class
}
```

Member Access and Inheritance

Although a subclass includes all of the members of its superclass, it cannot access those members ofthe superclass that have been declared as **private**.

A class member that has been declared as private will remain private to its class. It is not accessible by any code outside its class, including subclasses.

Program:

Output:

Conclusion:-

Questions-

1) **What is abstraction in java? How to achieve abstraction?**
2) **Can an abstract method be declared as static? Why?**
3) **What is the difference between Abstraction and Encapsulation?**
4) **What is difference between method overloading and method overriding?**