Programme Name: **BCS HONS**

Course Code: **CSC 2515**

Course Name: **Object Oriented Programming**

Open Book Examination

Date of Submission: **3/3/2020**

Submitted By:

Student Name**: Dipesh Tha Shrestha**

IUKL ID: **041902900028**

Semester**: Third Semester**

Intake**: September 2019**

Submitted To:

Faculty Name**: Som Prasad Shrestha**

Department**: LMS**

1. Answer the questions below based on the following class.

```
public class Coordinate
{
private int id;
private String name;
private double latitude;
private double longitude;
}
```

a. toString() method is very useful while printing state of any object. Write a toString() method for above class. Use all relevant information in class.

Answer: toString() is an in-built method in Java that returns the value given to it in string format. Hence, any object that this method is applied on, will then be returned as a string object. We can use toString() method to get string representation of an object. Whenever we try to print the Object reference then internally toString() method is invoked. If we did not define toString() method in your class then Object class toString() method is invoked otherwise our implemented/Overridden toString() method will be called.

```
public String toString(){
return "Coordinate [id="+id + ", name="+name + ",latitude="+latitude+",longitude="+longitude+"]";
}
Coordinate(int id, String name, double latitude, double longitude)
{
this.id=id;
this.name=name;
this.latitude=latitude;
this.longitude=longitude;
}
public static void main(String[] args) {
Coordinate obj= new Coordinate(1, "dipesh", 27.2046, 77.4977);
System.out.println(obj);
System.out.println(obj.toString());
} }
```

b. equals method is often used to check if two user defined objects are equal or not. Override equals method defined in Object class in above class that will return true if id is equal or latitude and longitude are equal, else false.

```
package javaTest;
public class Coordinate {
private int id;
private String name;
private double latitude;
```

```
 private double longitude;

Coordinate(int id, String name, double latitude, double longitude)
{
 this.id=id;
 this.name=name;
 this.latitude=latitude;
 this.longitude=longitude;
}
 @Override
 public boolean equals(Object o){
if(o == this){
 return true;
}
 if(!(o instanceof Coordinate)){
 return false;
 } Coordinate C =(Coordinate) o;
 Return
 Integer.compare(id,C.id)==0 && Double.compare(latitude,C.latitude)==0 &&
Double.compare(longitude, C.longitude) == 0;
 }
 public static void main(String[] args) {
Coordinate obj1= new Coordinate(1, "dipesh", 27.2046, 77.4977);
Coordinate obj2= new Coordinate(1," "dipesh", 27.2046, 77.4977);
if(obj1.equals(obj2)){
 System.out.println("Truel");
}else{ System.out.println("False");
 } } }
```

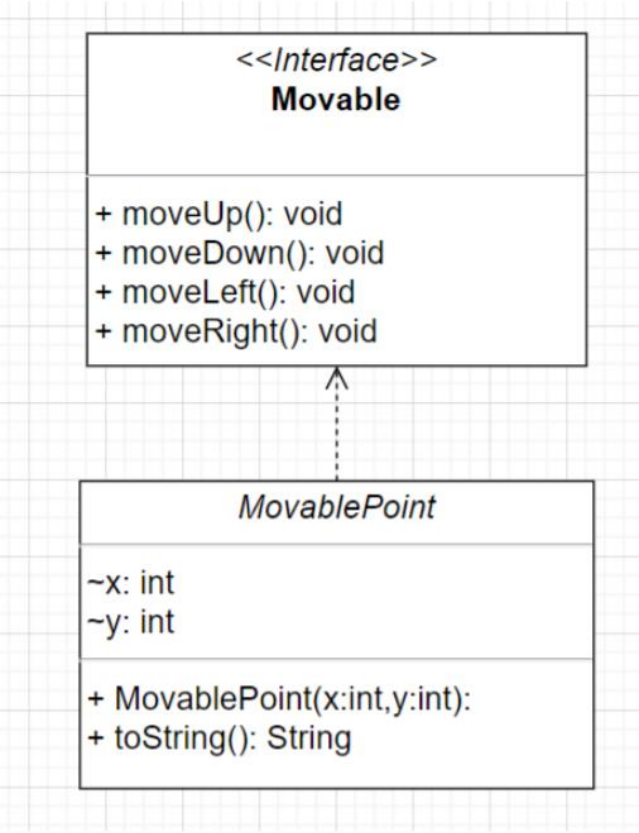
2. Consider a program that reads the record of employee. Write a program that reads input from a csv file and calculates the average salary. Complete the following main method code.

```
public static void main(String []args){
 File file=null;
FileReader reader=null;
 try{
file=new File("emp-data.csv");
 //complete code from here
 }catch(IOException ex)
{ ex.printStackTrace();
 } }
```

Solution:

```
 public static void main(String []args){
 File file=null;
FileReader reader=null;
 try{
 file=new File("emp-data.csv");
{
reader = new FileReader(new FileReader("F:\\CSVDemo.csv"));
String [] nextLine;
//read one line at a time
while ((nextLine = reader.readNext()) != null)
{
for(String token : nextLine)
{
System.out.println(token);
}
System.out.print("\n");
}
}
 }catch(IOException ex){
ex.printStackTrace();
 } }
```

3. Write code to realize following class diagram. Provide suitable implementation for the all required methods




```
interface Movable{
public void moveUp();
public void moveDown();
public void moveLeft();
public void moveRight();
}

public class MovablePoint implements Movable{

int x;
int y;
public void moveUp(){
}
public void moveDown(){
}
public void moveRight(){
}
public void moveLeft(){
}
public void movablePoint( int x, int y ){
```

```
}
public String toString(){
return null;
}
}
```

4. Write a Java program named "TwoLargestOddNumbers.java", which reads a sequence of positive integers from terminal, and terminates when a negative value is typed in. Right before it terminates, it should print out top 2 largest odd prime numbers (in descending order) ever seen in the sequence

```
package practiceFile;
import java.util.Scanner;
public class largeTwo {
@SuppressWarnings(&amp;quot;resource&amp;quot;)
public static void main(String[] args) {
int counter = 0;
Scanner in = new Scanner(System.in);
int top1 = -1;
int top2 = -1;
int num = in.nextInt();
while(num &amp;gt;= 0) {
if(num &amp;gt; top1) {
top2 = top1;
top1 = num;
}
else if(num &amp;gt; top2)
top2 = num;
num = in.nextInt();
counter++;
}
if(top1 &amp;gt;= 0 &amp;amp;&amp;amp; top2 &amp;gt;= 0) {
System.out.println(&amp;quot;Total numbers keyed in are: &amp;quot; + counter);
System.out.println(&amp;quot;The top 2 largest number are: &amp;quot;+top1 + &amp;quot;, &amp;quot; +

top2);
}
else if(top1 &amp;gt;= 0) {
System.out.println(&amp;quot;The top largest number is: &amp;quot;+top1);
}
}
}
```