**Week 7 Mon and Wed Lab Assignments**

* **Mark the following statements as True or False:**
* The constructor of a subclass specifies a call to the constructor of the superclass in the heading of the constructor’s definition. **False**
* The constructor of a subclass specifies a call to the constructor of the superclass using the name of the class.

**False**

* A subclass must define a constructor. **False**
* In Java, polymorphism is implemented using late binding. **True**
* **Consider the following class definition:**

public class AClass

{

private int u;

private int v;

public void print()

{

}

public void set(int x, int y)

{

}

public AClass()

{

}

public AClass(int x, int y)

{

}

}

**What is wrong with the following class definition?**

public class BClass AClass

{

private int w;

public void print()

{

System.out.println(“u + v + w = “ + (u+v+w);

}

public BClass()

{

super();

w = 0;

}

public BClass(int x, int y, int z)

{

super(x, y);

w = z;

}

}

**-Answers**

**-**public class BClass **extends** AClass

**-**set(x,y) instead of super(x,y)

* **Suppose that you have the following class definition:**

public class One

{

private int x;

private int y;

public void print()

{

System.out.println(x + “ “ + y);

}

protected void setDate(int u, int v)

{

x = u;

y = v;

}

}

**Consider the following class definition:**

public class Two extends One

{

private int z;

public void setDate(int a, int b, int c)

{

**//Postcondition: x = a; y =b; z = c;**

}

public void print()

{

**//Output the values of x, y and z**

}

}

* Write the definition of the method **setDate** of the **class** **Two** as described in the class definition.

-setDate(a,b), z = c;

* Write the definition of the method **print** of the **class Two** as described in the class definition.

-System.out.println(getx() + gety() + getz());

* **Explain and give example:** What does the operator **instanceof** do? - checks whether an object is an instance of a class or not.

// create an object of Main

Main obj = new Main();

// checks if obj is an instance of Main

boolean result2 = obj instanceof Main;

* **Given the ExceptionOriginal Class**, run this class for **Run 1** plugin values 12 and 5, **Run 2** plugin values 24 and 0, **Run 3** plugin values 2e and view the exceptions thrown, if any. ExceptionOriginal

Now modify the ExceptionOriginal class and place a test restricting user from dividing by o --- use if…. else block and display appropriate messages. **Run 1** plugin values 12 and 5, **Run 2** plugin values 24 and 0, and view the exceptions thrown, if any.

Now modify the ExceptionOriginal class and place a test restricting user from dividing by o (ArithmeticException class should be used to catch this exception) and other exception to be caught is (InputMismatchException)--- using try…catch block and display appropriate messages by the exception classes. **Run 1** plugin values 45 and 2, **Run 2** plugin values 18 and 0, **Run 3** plugin values 2753 and 2f1rl.