**3.Ans**

**[1,3,5]:**

**package** asse;

**import** java.lang.reflect.\*;

**public** **class** test13

{

**static** Class *C*;

**public** **static** **void** methodsName(String s) **throws** ClassNotFoundException {

*C* = Class.*forName*(s);

Method[] m = *C*.getDeclaredMethods();

**if** (m.length == 0)

System.***out***.println("no methods");

**else** {

**for** (**int** i = 0; i < m.length; i++) {

Method method = m[i];

System.***out***.println("Method Name:- " + method.getName());

System.***out***.println("Declared Class :- " + method.getDeclaringClass());

Class p[] = method.getParameterTypes();

**for** (**int** j = 0; j < p.length; j++) {

System.***out***.println("\tParameters :- " + p[j].getName());

}

System.***out***.println("Return Type :- " + method.getReturnType());

**int** mm = method.getModifiers();

System.***out***.println("Modifiers :- " + Modifier.*toString*(mm));

System.***out***.println("---------------------------------------");

}

}

}

**public** **static** **void** attributesDetails(String s) **throws** ClassNotFoundException {

*C* = Class.*forName*(s);

Field[] f = *C*.getDeclaredFields();

**if** (f.length == 0)

System.***out***.println("no atttributes");

**else** {

**for** (**int** i = 0; i < f.length; i++) {

Field field = f[i];

System.***out***.println("Fields Name:- " + field.getName());

System.***out***.println("Declared Class :- " + field.getDeclaringClass());

System.***out***.println("DataType:- " + field.getType());

System.***out***.println("------------------------------");

}

}

}

@SuppressWarnings("deprecation")

**public** **static** Object instanceCreation(String s) **throws** ClassNotFoundException {

**try** {

**return** Class.*forName*(s).~~newInstance~~();

} **catch** (Exception E) {

System.***out***.println("");

}

System.***out***.println("Unable to Instance");

**return** **null**;

}

**public** **static** **void** constructersDetails(String s) **throws** ClassNotFoundException {

*C* = Class.*forName*(s);

Constructor[] constructors = *C*.getDeclaredConstructors();

**for** (Constructor cons : constructors) {

System.***out***.println("Constructor Name: " + cons.getName());

**int** modifier = cons.getModifiers();

String mod = Modifier.*toString*(modifier);

System.***out***.println("Modifier: " + mod);

System.***out***.println("Parameters: " + cons.getParameterCount());

Class p[] = cons.getParameterTypes();

**if** (p.length == 0) {

System.***out***.println("\t0 parameters Constructor ");

System.***out***.println("-------------------------------");

} **else** {

**for** (**int** j = 0; j < p.length; j++) {

System.***out***.println("\tParameters :- " + p[j].getName());

}

System.***out***.println("------------------------------");

}

}

}

**public** **static** **void** main(String[] args) **throws** ClassNotFoundException {

**for** (**int** i = 0; i < args.length; i++) {

System.***out***.println("------------------------------");

//Methods Details");

System.***out***.println("-------------------------------");

*methodsName*(args[i]);

//End Of Method

System.***out***.println("\n-----------------------------");

//Attributes Details

System.***out***.println("-------------------------------");

*attributesDetails*(args[i]);

// End Of Method

System.***out***.println("\n------------------------------");

//Constructors Details

System.***out***.println("--------------------------------");

*constructersDetails*(args[i]);

// End Of Method

System.***out***.println("\n------------------------------");

//Instance Details

System.***out***.println("--------------------------------");

Object O = *instanceCreation*(args[i]);

System.***out***.println(O.hashCode());

System.***out***.println(O.toString());

// End Of Method

}

}

}

**[2]:**

class Generic<T> {

T t;

int a, b;

Generic() {

}

Generic(T t, int a) {

this.t = t;

this.a = a;

}

Generic(T t, int a, int b) {

this.t = t;

this.a = a;

this.b = b;

}

public void get() {

System.out.println(t + " " + a + " " + b);

}

}

**[4]:**

class abc {

private int a;

int b, c;

public abc(int r) {

b = r;

}

public abc(int r, int s) {

b = r;

c = s;

}

public abc(int a, int b, int c) {

this.a = a;

this.b = b;

this.c = c;

}

public void alfa() {

a = 10;

b = 20;

c = 30;

}

}