Project -1:

Writing a program in Java to implement implicit and explicit type casting

**package** Project;

**public** **class** Project1 {

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

**byte** a=10;

**short** b= a; //implicit type casting

System.***out***.println("a = "+a);

System.***out***.println("b = "+b);

**short** c=20;

**byte** d =(**byte**)c; //explicit type casting

System.***out***.println("c = "+c);

System.***out***.println("d = "+d);

}

}

Project -2:

Writing a program in Java to implement access modifiers

**class** defAccessSpecifier

{

**void** display()

{

System.***out***.println("You are using defalut access specifier");

}

}

**public** **class** accessSpecifiers1 {

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

System.***out***.println("Default Access Specifier");

defAccessSpecifier obj = **new** defAccessSpecifier();

obj.display();

}

}

**class** priaccessspecifier

{

**private** **void** display()

{

System.***out***.println("You are using private access specifier");

}

}

**public** **class** accessSpecifiers2 {

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

System.***out***.println("Private Access Specifier");

priaccessspecifier obj = **new** priaccessspecifier();

obj.display();

}

}

**package** pack1;

**public** **class** proaccessspecifiers {

**protected** **void** display()

{

System.***out***.println("This is protected access specifier");

}

}

**package** pack2;

**import** pack1.\*;

**public** **class** accessSpecifiers3 **extends** proaccessspecifiers {

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

accessSpecifiers3 obj = **new** accessSpecifiers3 ();

obj.display();

}

}

**package** pack1;

**public** **class** pubaccessspecifiers {

**public** **void** display()

{

System.***out***.println("This is Public Access Specifiers");

}

}

**package** pack2;

**import** pack1.\*;

**public** **class** accessSpecifiers4 {

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

pubaccessspecifiers obj = **new** pubaccessspecifiers();

obj.display();

}

}

Project -3:

Writing a program in Java to verify implementations of methods and ways of calling a method

**public** **class** Method {

**public** **int** Adding(**int** a,**int** b) {

**int** c=a+b;

**return** c;

}

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

Method m=**new** Method();

**int** s= m.Adding(20,5);

System.***out***.println("Addition is :"+s);

}

}

**public** **class** callbyvalue {

**int** a=150;

**int** addition(**int** a) {

a =a+100;

**return**(a);

}

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

callbyvalue c = **new** callbyvalue();

System.***out***.println("Before Addition value of data is "+c.a);

c.addition(100);

System.***out***.println("After Addition value of data is "+c.a);

}

}

**package** Project3;

**public** **class** MethodOverLoading {

**public** **void** area(**int** b,**int** h){

System.***out***.println("Area of Triangle : "+(0.5\*b\*h));

}

**public** **void** area(**int** r) {

System.***out***.println("Area of Circle : "+(3.14\*r\*r));

}

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

MethodOverLoading ob=**new** MethodOverLoading();

ob.area(15,20);

ob.area(10);

}

}

Project -4:

Writing a program in Java to verify the implementations of constructor types

**package** Project4;

**class** Vehicle{

**int** id;

String name;

Vehicle(**int** i,String n)

{

id=i;

name=n;

}

**void** display() {

System.***out***.println(id+" "+name);

}

}

**public** **class** Constructor1 {

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

Vehicle v1=**new** Vehicle(252,"Innova");

Vehicle v2=**new** Vehicle(101,"BMW");

v1.display();

v2.display();

}

}

**package** Project4;

**class** Car{

**int** id;

String name;

**void** display() {

System.***out***.println(id+" "+name);

}

}

**public** **class** Constructors {

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

Car c1=**new** Car();

Car c2=**new** Car();

c1.display();

c2.display();

}

}

Project -5:

Writing a program in Java to verify implementations of collections

**package** Project;

**import** java.util.\*;

**public** **class** Collections {

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

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

ArrayList<String> c=**new** ArrayList<String>();

c.add("Hyderabad");//

c.add("Pune");

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

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

Vector<Integer> v = **new** Vector();

v.addElement(25);

v.addElement(45);

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

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

LinkedList<String> names=**new** LinkedList<String>();

names.add("Revathi");

Iterator<String> itr=names.iterator();

**while**(itr.hasNext()){

System.***out***.println(itr.next());

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

HashSet<Integer> s1=**new** HashSet<Integer>();

s1.add(321);

s1.add(284);

s1.add(402);

s1.add(581);

System.***out***.println(s1);

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

LinkedHashSet<Integer> s2=**new** LinkedHashSet<Integer>();

s2.add(1);

s2.add(2);

s2.add(3);

s2.add(4);

System.***out***.println(s2);

}

}

}

Project -6:

Writing a program in Java to verify implementations of maps

**package** Project;

**import** java.util.\*;

**public** **class** Maps {

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

HashMap<Integer,String> h=**new** HashMap<Integer,String>();

h.put(1,"Reva");

h.put(2,"Shiva");

h.put(3,"Ramya");

System.***out***.println("\nThe elements of Hashmap are ");

**for**(Map.Entry m:h.entrySet()){

System.***out***.println(m.getKey()+" "+m.getValue());

}

Hashtable<Integer,String> t=**new** Hashtable<Integer,String>();

t.put(4,"kiran");

t.put(5,"karan");

t.put(6,"Ram");

t.put(7,"Lakshman");

System.***out***.println("\nThe elements of HashTable are ");

**for**(Map.Entry n:t.entrySet()){

System.***out***.println(n.getKey()+" "+n.getValue());

}

TreeMap<Integer,String> tm=**new** TreeMap<Integer,String>();

tm.put(8,"Tom");

tm.put(9,"Meera");

tm.put(10,"John");

System.***out***.println("\nThe elements of TreeMap are ");

**for**(Map.Entry l:tm.entrySet()){

System.***out***.println(l.getKey()+" "+l.getValue());

}

}

}

Project -7:

Writing a program in Java to verify the implementation of inner classes

**package** Project;

**public** **class** Class1 {

**private** String m="Hello";

**class** Inner{

**void** hello(){System.***out***.println(m+", Inner Classes");}

}

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

Class1 o1=**new** Class1();

Class1.Inner in=o1.**new** Inner();

in.hello();

}

}

**public** **class** Class2 {

**private** String m="Inner Classes";

**void** display(){

**class** Inner{

**void** msg(){

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

}

}

Inner l=**new** Inner();

l.msg();

}

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

Class2 o2=**new** Class2 ();

o2.display();

}

}

**abstract** **class** Anonymous {

**public** **abstract** **void** display();

}

**public** **class** Class3 {

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

Anonymous i = **new** Anonymous() {

**public** **void** display() {

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

}

}

i.display();

}

}

Project -8:

Write a program to create strings and display the conversion of string to StringBuffer and StringBuilder.

**package** Project;

**public** **class** StringProgram {

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

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

String sl=**new** String("Hello ");

System.***out***.println(sl.length());

String sub=**new** String("Everyone");

System.***out***.println(sub.substring(2));

String s1="Hello";

String s2="Heldo";

System.***out***.println(s1.compareTo(s2));

String s4="";

System.***out***.println(s4.isEmpty());

String s5="Hello";

System.***out***.println(s1.toLowerCase());

String s6="Heldo";

String replace=s2.replace('d', 'l');

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

String a="Welcome Everyone";

String b="WeLcOmE Everyone";

System.***out***.println(a.equals(b));

System.***out***.println("\nCreating StringBuffer");

StringBuffer s=**new** StringBuffer("Welcome to all!");

s.append("Have a nice Day");

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

s.insert(0, 'w');

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

StringBuffer sb=**new** StringBuffer("Hello");

sb.replace(0, 2, "hEl");

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

sb.delete(0, 1);

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

System.***out***.println("\nCreating StringBuilder");

StringBuilder sb1=**new** StringBuilder("Have a nice");

sb1.append(" Day");

System.***out***.println(sb1);

System.***out***.println(sb1.delete(0, 1));

System.***out***.println(sb1.insert(1, "Welcome"));

System.***out***.println(sb1.reverse());

System.***out***.println("\nConversion of Strings to StringBuffer and StringBuilder");

String str = "Hello";

StringBuffer sbr = **new** StringBuffer(str);

sbr.reverse();

System.***out***.println("String to StringBuffer");

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

StringBuilder sbl = **new** StringBuilder(str);

sbl.append("world");

System.***out***.println("String to StringBuilder");

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

}

}

Project -9:

Writing a program in Java to verify implementation of arrays

**package** Project;

**public** **class** Arrays {

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

**int** arr[]= {1,2,3,4,5,6,7,8,9,10};

**for**(**int** i=0;i<10;i++) {

System.***out***.println("Elements of array arr : "+arr[i]);

}

**int**[][] mul = {{1,2,3,4}, { 5,6,7} };

System.***out***.println("\nLength of row 1: " + mul[0].length);

System.***out***.println("\nLength of row 2: " + mul[1].length);

}

}

Project -10:

Writing a program in Java to verify implementations of regular expressions

**package** Project;

**import** java.util.regex.\*;

**public** **class** RegularExpressions {

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

String s1 = "[a-z]+";

String s2 = "Regular Expressions";

Pattern p =Pattern.*compile*(s1);

Matcher m = p.matcher(s2);

**while** (m.find())

System.***out***.println( s2.substring( m.start(), m.end() ) );

}

}