**Assignment 8:**

**Group1:**

**What output is produced by the following code?**

1.

public class inheritance1 {  
 public static void main (String[] args) {  
 Sub obj = new Sub();  
 obj.display();  
 }  
}  
  
class Super {  
 int num = 10;  
 int id = 20;  
}  
  
class Sub extends Super {  
 int num = 100;  
 public void display() {  
 System.out.println(num);  
 System.out.println(super.num);  
 System.out.println(id);  
 }  
}

2.

class parent {  
 public void display() {  
 System.out.println("partent display(); ");  
 }  
}  
  
class child extends parent {  
 @Override  
 public void display () {  
 System.out.println("child display(); ");  
 }  
 public void test() {  
 display();  
 }  
}  
  
public class overriding1 {  
 public static void main(String[] args) {  
 child obj = new child();  
 obj.display();  
 obj.test();  
 }  
}

3.

class parent2 {  
 public void display() {  
 System.out.println("partent display(); ");  
 }  
}  
  
class child2 extends parent2 {  
 @Override  
 public void display() {  
 super.display();  
 System.out.println("child display(); ");  
 }  
 public void test() {  
 display();  
 super.display();  
 }  
}  
  
public class overriding2 {  
 public static void main(String[] args) {  
 child2 obj = new child2();  
 obj.test();  
 obj.display();  
 }  
}

4.

class superCon {

superCon(){  
 System.out.println("super constructor");  
 }  
 }  
   
 class subCon extends superCon{  
 subCon(){  
 System.out.println("sub constructor");  
 }  
 }  
   
 public class constrctor1 {  
 public static void main(String[] args) {  
 subCon obj = new subCon();  
 }  
}

5.

class ParentClass2 {  
 public ParentClass2() {}   
 public void method1() {  
 System.out.println("ParentClass method1()"); }  
 public void method2() {  
 System.out.println("ParentClass method2()");}  
}   
  
class ChildClass2 extends ParentClass2 {  
 public ChildClass2() { }  
  
 @Override  
 public void method1() {  
 System.out.println("ChildClass method1()"); }  
   
 public void method3(){  
 System.out.println("ChildClass method3()"); }  
}  
  
public class constructor2 {  
 public static void main(String[] args) {  
 ChildClass2 child = new ChildClass2();  
 child.method1();  
 child.method2();  
 child.method3();}  
}

6.

class ParentClass3 {  
 public ParentClass3() {  
 System.out.println("Parent Class."); }  
   
 public void method1() {  
 System.out.println("Parent Class : Method1()."); }  
}  
  
class ChildClass3 extends ParentClass3 {  
 public ChildClass3() {  
 System.out.println("Child Class.");}  
   
 @Override  
 public void method1() {  
 super.method1();  
 System.out.println("Child Class : Method1().");}  
}  
  
public class constructor3 {  
 public static void main(String[] args) {  
 ChildClass3 child = new ChildClass3();  
 child.method1();}  
}

**Group 2:**

**1. What output is produced by the following code?**

class Car{

String Color;

int door;

void drive(){

System.out.println( "drive, Car" ); }

void stop(){

System.out.println( "stop, Car" );}

}

class FireEngine extends Car {

void drive(){

System.out.println( "drive, FireEngine" );}

}

class Ambulance extends Car{

void stop(){

System.out.println( "stop, Ambulance" );}

}

public class poly3{

public static void main( String[] args ){

Car c1 = new FireEngine();

Car c2 = new Ambulance();

c1.drive();

c1.stop();

c2.drive();

c2.stop();

}

}

**2. What output is produced by the following code?**

class A{

int i = 10;

public String x(){return "A.x";}

}

class B extends A{

int i = 5;

public String x(){return "B.x";}

public String y(){return "y";}

}

class B2 extends A{

int i = 3;

public String x(){return "B2.x";}

}

public class Poly4 {

public static void main(String[] args) {

A obj = new B();

A obj2 = new B2();

System.out.println(obj.x());

System.out.println(obj2.x());

System.out.println(obj.i);

System.out.println(obj2.i);

}

}

**3. Complete Java program to produce the following output.**

class Vehicle {

String name;

String company;

int price;

public Vehicle(String name, String company, int price) {

this.name = name;

this.company = company;

this.price = price; }

public void printInfo() {

}

}

class Car extends Vehicle {

int passengers;

Car(String name, String company, int price, int passengers) {

super(name, company, price);

this.passengers = passengers; }

@Override

public void printInfo() {

}

}

class Truck extends Vehicle {

int maxLoad;

@Override

public void printInfo() {

}

}

public class PolymorphismDemo {

public static void main(String[] args) {

Vehicle v1 = new Car("Sonata", "Hyundai", 3000, 5);

Vehicle v2 = new Truck("Ranger", "Ford", 3500, 6);

v1.printInfo();

v2.printInfo(); }

}

//output

car type: Sonata Maker: Hyundai Price: 3000 Passengers: 5

car type: Ranger Maker: Ford Price: 3500 Max Load: 6

**Group 3:**

**1. Complete Java program to produce the following output.**

interface Printable{   
 void print(); }

interface Showable{   
 void show(); }   
   
class p3\_1 implements Printable,Showable{

public static void main(String args[]){   
 p3\_1 obj = new p3\_1();   
 obj.print();   
 obj.show(); }

}

//output

Hello  
Welcome

**2. Complete Java program to produce the following output.**

interface printable{   
void print();   
}

//output

Hello

**3. What output is produced by the following code?**

interface Printable{   
 void print(); }

interface Showable extends Printable{   
 void show(); }

class p3\_3 implements Showable{   
 public void print(){System.out.println("Hello");}   
 public void show(){System.out.println("Welcome");}   
 public static void main(String args[]){   
 p3\_3 obj = new p3\_3();   
 obj.print();   
 obj.show(); }   
}

**4. Complete Java program to produce the following output.**

// Example of Method Overloading

class p3\_4{

public static void main(String args[]){   
 p3\_4 obj=new p3\_4();   
 obj.sum(10.5,10.5);   
 obj.sum(20,20);   
 }   
}

//output

21.0  
40

**5. What output is produced by the following code?**

class Animal{   
void eat(){System.out.println("animal is eating...");}   
}   
   
class Dog extends Animal{   
void eat(){System.out.println("dog is eating...");}   
}   
   
class p3\_5 extends Dog{   
public static void main(String args[]){   
Animal a=new p3\_5();   
a.eat();   
}

}

**6. What output is produced by the following code?**

abstract class Bike{   
 Bike(){System.out.println("bike is created");}   
 abstract void run();   
 void changeGear(){System.out.println("gear changed");}   
 }   
   
 class Honda extends Bike{   
 void run(){System.out.println("running safely..");}   
 }   
   
 class p3\_6{   
 public static void main(String args[]){   
 Bike obj = new Honda();   
 obj.run();   
 obj.changeGear();   
 }   
}

**Group 4:**

**1. Complete Java program to produce the following output.**

import java.util.\*;  
public class ex6\_1ItrRemoveElement {  
  
public static void main(String[] args){  
 String removeElem = "apple";  
 List<String> myList = new ArrayList<String>();  
 myList.add("banana");

System.out.print("Before remove:");  
 System.out.println(myList);  
   
 Iterator<String> itr = myList.iterator();  
 while( ){  
 if( ){  
 itr.remove();  
 }  
 }  
 System.out.print("After remove:");  
 System.out.println(myList);  
 }  
}

// ouput

Before remove: [banana, apple, mango, orange, cherry]

After remove: [banana, mango, orange, cherry]

**2. Complete Java program to produce the following output.**

import java.util.\*;  
public class ex6\_2iterateHashMap {  
 public static void main(String[] args) {  
 HashMap< Integer, String> hMap = new HashMap<Integer, String>();  
  
 hMap.put( , "Bellevue");  
 hMap.put( , "College");  
 hMap.put( , "CS211");

Iterator itr = cl.iterator();  
 while (itr.hasNext()) {  
 System.out.println(itr.next());  
 }  
 }  
}

//output

Bellevue  
CS211  
College

**3. What output is produced by the following code?**

List list = Arrays.asList("one Two three Four five one three Four".split(" "));  
 System.out.println("List :"+list);  
 List sublist = Arrays.asList("three Four".split(" "));  
 System.out.println("SubList :"+sublist);  
 System.out.println("indexOfSubList: " + Collections.indexOfSubList(list, sublist));  
 System.out.println("lastIndexOfSubList: " + Collections.lastIndexOfSubList(list, sublist));  
   
 List sublist2 = Arrays.asList("Two Four".split(" "));  
 System.out.println("SubList :"+sublist2);  
 System.out.println("indexOfSubList2: " + Collections.indexOfSubList(list, sublist2));

**4. What output is produced by the following code?**

List list = Arrays.asList("1 2 3 4 5".split(" "));

System.out.println("List :"+list);

Collections.rotate(list, 2);

System.out.println("rotate: " + list);

Collections.rotate(list, 4);

System.out.println("rotate: " + list);

Collections.rotate(list, 3);

System.out.println("rotate: " + list);

**5. What output is produced by the following code?**

ArrayList<Integer> al = new ArrayList<Integer>();

HashSet<Integer> hs = new HashSet<Integer>();

for (int i= 0; i<5; i++) {

al.add(i);

hs.add(i);

hs.add(i\*2);

}

System.out.println("array");

Iterator ai = al.iterator();

while(ai.hasNext()){

System.out.println(ai.next());

}

Iterator hi = hs.iterator();

System.out.println("hashset");

while(hi.hasNext()){

System.out.println(hi.next());

}

**6. Complete Java code to produce the following output.**

public static void main(String[] args) {

// ArrayList

List a1 = new ArrayList();

// LinkedList

// HashSet

// HashMap

}

//output

ArrayList: [Seattle, Bellevue]

LinkedList: [Seattle, Bellevue]

HashSet: [Seattle, Bellevue]

HashMap: {Seattle=1, Yakima=2, Bellevue=3}

**Group 5:**

**1. What output is produced by the following code?**

public class forLoopType2{  
 public static void main (String[] args) {  
 int myArray[] = {5,4,3,2,1,0};  
 for (int num : myArray){  
 System.out.println(num % 2);  
 }  
 }

**2. What is the output of this program?**  
import java.util.\*;  
 class hashtableQ1 {  
 public static void main(String args[]) {  
 Hashtable obj = new Hashtable();  
 obj.put("A", new Integer(3));  
 obj.put("B", new Integer(2));  
 obj.put("C", new Integer(8));  
 System.out.print(obj.contains(new Integer(5)));  
 }  
 }

**3. What is the output of this program?**  
import java.util.\*;  
 class hashtableQ1 {  
 public static void main(String args[]) {  
 Hashtable obj = new Hashtable();  
 obj.put("A", new Integer(3));  
 obj.put("B", new Integer(2));  
 obj.put("C", new Integer(8));  
 System.out.print(obj.contains(new Integer(5)));  
 }  
 }

**4. What is the output of this program?**  
import java.util.\*;  
 class hashtableQ3 {  
 public static void main(String args[]) {  
 Hashtable obj = new Hashtable();  
 obj.put("A", new Integer(3));  
 obj.put("B", new Integer(2));  
 obj.put("C", new Integer(8));  
 obj.remove(new String("A"));  
 System.out.print(obj);  
 }  
 }

**5. What is the output of this program?**  
import java.util.\*;  
 class hashtableQ3 {  
 public static void main(String args[]) {  
 Hashtable obj = new Hashtable();  
 obj.put("A", new Integer(3));  
 obj.put("B", new Integer(2));  
 obj.put("C", new Integer(8));  
 obj.remove(new String("A"));  
 System.out.print(obj);  
 }  
 }

**Group 6:**

**1. Given the output stream A,B,C,D,E,F Write down the sequence of operations (Push for stack and Pop for unstack) which would produce the sequence C,B,D,E,F,A**

**2. What is the output of this program?**

mport java.util.\*;

public class LinkedListExample {

public static void main(String args[]) {

/\* Linked List Declaration \*/

LinkedList<String> linkedlist = new LinkedList<String>();

/\*add(String Element) is used for adding

\* the elements to the linked list\*/

linkedlist.add("Item1");

linkedlist.add("Item5");

linkedlist.add("Item3");

linkedlist.add("Item6");

linkedlist.add("Item2");

/\*Display Linked List Content\*/

System.out.println("Linked List Content: " +linkedlist);

/\*Add First and Last Element\*/

linkedlist.addFirst("First Item");

linkedlist.addLast("Last Item");

System.out.println("LinkedList Content after addition: " +linkedlist);

/\*This is how to get and set Values\*/

Object firstvar = linkedlist.get(0);

System.out.println("First element: " +firstvar);

linkedlist.set(0, "Changed first item");

Object firstvar2 = linkedlist.get(0);

System.out.println("First element after update by set method: " +firstvar2);

/\*Remove first and last element\*/

linkedlist.removeFirst();

linkedlist.removeLast();

System.out.println("LinkedList after deletion of first and last element: " +linkedlist);

/\* Add to a Position and remove from a position\*/

linkedlist.add(0, "Newly added item");

linkedlist.remove(2);

System.out.println("Final Content: " +linkedlist);

}

}

**3. What is the running time of each the method?**

public int add100(int[] array) {

if (array.length < 100) {

return 0;

}

int sum = 0;

for (int i = 0; i < 100; i++) {

sum += array[i];

}

return sum;

}

public static void m4(int[] arr) { for (int i=0; i<arr.length; i++) {    System.out.println(arr[i] \* 10); } for (int j=arr.length-1; j>=0; j--) {    System.out.println(arr[j] / 10); }}

public static void m5(int[] arr) { for (int i=0; i<15; i++) {   for (int j=0; j<arr.length; j++) {     System.out.println(Math.max(arr[i], arr[j])); }}}

**4. Using big-O notation in terms of the parameter n, how much time does the following method take?**

public static int mystery(int n) {

int count = 0;

int cur = 1;

while (cur < n) {

count++;

cur = cur \* 2;

}

return cur;

}

**5. Each of the below methods determine m, n without using Math.pow. Using big-O notation in terms of n, how much time does each take?**

a.

int pow(int m, int n) {

int ret = 1;

for (int i = 0; i < n; i++) {

ret \*= m;

}

return ret;

}

b.

int pow(int m, int n) {

int ret = 1;

int k = m;

int i = n;

while (i > 0) {

if (i % 2 == 1) ret \*= k;

k \*= k;

i /= 2;

}

return ret;

}

**6. Why is binary search O(log2 N) ?**

**Group 7:**

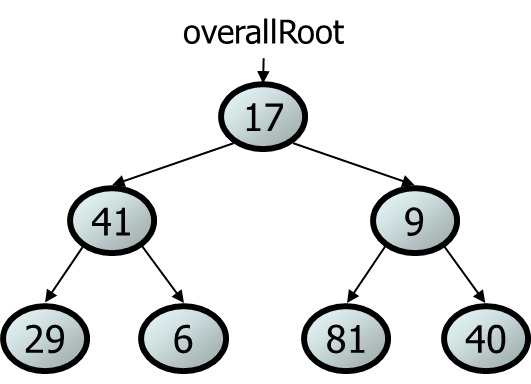
**1. Create binary search tree and fill out the following array for {“apple”, “banana”, “cherry”, “date”, grape”, “mango”}**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Left Node | Data | Right Node |
| A[0] |  |  |  |
| A[1] |  |  |  |
| A[2] |  |  |  |
| A[3] |  |  |  |
| A[4] |  |  |  |
| A[5] |  |  |  |

**2. Create min heap tree for {7, 3, 5, 2, 4, 6}**

**3. When the root is removed from the above heap tree, how do we alter the tree? Create a new heap tree step by step.**

**4. Find the following orderings for traversals.**



Pre-order:

In-order:

Post-order