|  |  |
| --- | --- |
| Course- BTech | Type- Core |
| Course Code- **CSET** | Course Name- **Object Oriented Programming Using Java** |
| Year- First | Semester- Even Batch- BTech 2nd Semester |

**Tutorial-5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tutorial No.** | **Name** | **CO1** | **CO2** | **CO3** |
| **1** | **Basics** |  | **--** | **--** |

**Objective:** The main objective of this tutorial is to learn about the functions basics of Java language.

Q1. What is output of below code?

public

class Main {

public

int foo() { return 10; }

public

char foo() { return 'a'; }

public

static void main(String args[]) {

foo();

}

}

Ans: Functions cannot be overloaded only on basis of return type so will give compile error.

Q2. What is the following code doing, what will be its output?

class Test {

// Returns an array such that first element

// of array is a+b, and second element is a-b

static int[] getSumAndSub(int a, int b)

{

int[] ans = new int[2];

ans[0] = a + b;

ans[1] = a - b;

// returning array of elements

return ans;

}

// Driver method

public static void main(String[] args)

{

int[] ans = getSumAndSub(100, 50);

System.out.println("Sum = " + ans[0]);

System.out.println("Sub = " + ans[1]);

}

}

Ans: Array used to return 2 variables so output will be

150

50

Q3. What is need of 2nd class: MultiDivAdd in this code and what will be the output?

class MultiDivAdd {

int mul; // To store multiplication

double div; // To store division

int add; // To store addition

MultiDivAdd(int m, double d, int a)

{

mul = m;

div = d;

add = a;

}

}

class Test {

static MultiDivAdd getMultDivAdd(int a, int b)

{

// Returning multiple values of different

// types by returning an object

return new MultiDivAdd(a \* b, (double)a / b, (a + b));

}

// Driver code

public static void main(String[] args)

{

MultiDivAdd ans = getMultDivAdd(10, 20);

System.out.println("Multiplication = " + ans.mul);

System.out.println("Division = " + ans.div);

System.out.println("Addition = " + ans.add);

}

}

Ans: Class MultiDivAdd used to return 3 values, so output:

200

0.5

30

Q4. What is use of MyInt class and what will be its output?

class MyInt

{

int val;

public MyInt(int x)

{

val = x;

}

public void setValue(int newval)

{

val = newval;

}

public int getValue()

{

return val;

}

}

public class Main

{

public static void swap(Integer i, Integer j) {

Integer temp = new Integer(i);

i = j;

j = temp;

}

public static void swap1(MyInt i, MyInt j) {

MyInt temp = new MyInt(i.getValue());

i.setValue(j.getValue());

j.setValue(temp.getValue());

}

public static void main(String[] args) {

Integer i = new Integer(10);

Integer j = new Integer(20);

swap(i, j);

System.out.println("i = " + i + ", j = " + j);

MyInt i1 = new MyInt(10);

MyInt j1 = new MyInt(20);

swap1(i1, j1);

System.out.println("i = " + i1.getValue() + ", j = " + j1.getValue());

}

}

Ans: MyInt class helps to get the modified value outside main also, so output:

i = 10,j = 20

I = 20,j = 10

Q5 What is below concept called for using same function name for different arguments, what is its output?

public class Main

{

public int sum(int x, int y) { return (x + y); }

// Overloaded sum(). This sum takes three int parameters

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

{

return (x + y + z);

}

// Overloaded sum(). This sum takes two double

// parameters

public double sum(double x, double y)

{

return (x + y);

}

// Driver code

public static void main(String args[])

{

Main s = new Main();

System.out.println(s.sum(10, 20));

System.out.println(s.sum(10, 20, 30));

System.out.println(s.sum(10.5, 20.5));

}

}

Ans: this is for method overloading. Output:

30

60

31.0

Q6 What is effect of below code, what will be the output?

class Base {

public final void foo() {System.out.println("Base");}

}

class Derived extends Base {

public void foo() { System.out.println("Derived"); }

}

public class Main {

public static void main(String args[]) {

Base b = new Derived();

b.foo();

}

}

Ans: the final keyword will not allow the same method in derived class so compile error.

Q7 Check the difference in below code from Q6 and what will be output?

class Base {

private void foo() {System.out.println("Base");}

}

class Derived extends Base {

public void foo() { System.out.println("Derived"); }

}

public class Main {

public static void main(String args[]) {

Base b = new Derived();

b.foo();

}

}

Ans: since foo() is private in base class so cannot be derived, compile error

Q8 What is the below code doing?

public class Main {

public int Power(int x, int n)

{

if(n == 0)

return 1;

if(n%2 == 1)

{

int y = Power(x, (n-1)/2);

return x\*y\*y;

}

else

{

int y = Power(x, n/2);

return y\*y;

}

}

public static void main(String args[]) {

Main pow = new Main();

System.out.println(pow.Power(2,5));

System.out.println(pow.Power(2,6));

}

}

Ans: The code recursively calculates power of any number, output:

32

64

Q9 What is effect of below code, show its output?

public class Main {

public static void Reverse(int arr[], int i, int j)

{

if(i < j && i < arr.length && j >= 0)

{

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

Reverse(arr, i+1, j-1);

}

}

public static void main(String args[]) {

int arr[] = new int []{3,7,1,10,2,6,12};

for(int i=0; i < arr.length; i++)

System.out.print(arr[i] + " ");

System.out.println("");

Reverse(arr, 0, arr.length-1);

for(int i=0; i < arr.length; i++)

System.out.print(arr[i] + " ");

}

}

Ans: the code reverses the array using recursive function, output:

3 7 1 10 2 6 12

12 6 2 10 1 7 3

Q10 What will be output of below code?

public class Main {

// Normal main()

public static void main(String[] BU) {

System.out.println("Hi BU (from main)");

Main.main("BU");

}

// Overloaded main methods

public static void main(String arg1) {

System.out.println("Hi, " + arg1);

Main.main("Dear BU","My BU");

}

public static void main(String arg1, String arg2) {

System.out.println("Hi, " + arg1 + ", " + arg2);

}

}

Ans: it show how to overload main method, the output will be:

Hi BU (from main)

Hi, BU

Hi, Dear BU, My BU