Java Assignment -1

1. 10 differences between compiler and interpreter.

|  |  |
| --- | --- |
| Compiler | Interpreter |
| Scans the entire program and translates it as a whole into machine | Translates program one statement at a time. |
| Compilers usually take a large amount of time to analyse the source code. | Interpreters usually take less amount of time to analyze the source code. |
| The overall execution time is comparatively faster than interpreters. | The overall execution time is comparatively slower than compilers. |
| Generates Object Code which further requires linking, hence requires more memory. | No Object Code is generated, hence are memory efficient. |
| Store machine language as machine code on the disk | Not saving machine code at all. |
| Generates output program (in the form of exe) which can be run independently from the original program. | Do not generate output program. So they evaluate the source program at every time during execution. |
| The program code is already translated into machine code. Thus, it code execution time is less. | Interpreters are easier to use, especially for beginners. |
| It is best suited for the Production Environment | It is best suited for the program and development environment. |
| Program execution is separate from the compilation. It performed only after the entire output program is compiled. | Program Execution is a part of Interpretation process, so it is performed line by line. |
| C,C++,C#, Scala, Java all use complier. | PHP, Perl, Ruby uses an interpreter. |

2.Give details and 5 examples of Strongly typed and loosely typed language?

Strongly Typed: A **strongly typed** language on the contrary wants types specified. **strongly-typed** languages don’t allow implicit conversions between unrelated types.

A lot of — but not all — developers agree that the essence of a strongly typed language is the fact that it converts a variable or value’s type to suit the current situation automatically. This means the “123” is always treated as a string and is never used as a number without manual conversion or intervention.

* temp=”Hello World!”

temp= temp+10;

//program terminates with error.

* x=1;

y=”2”;

z=x+y;

* var=21;

var=var+”dot”;

print(var);

TypeError: unsupported operand type(s) for +: ‘int’ and ‘str’

Weakly Typed: In programming we call a language **loosely typed** when you don’t have to explicitly specify types of variables and objects.

**Weakly-typed** languages make conversions between unrelated types implicitly

The interpreter or compiler attempts to make the best of what it is given by using variables in ways that might seem confusing at first, but make sense once we understand what they are doing.

* $temp=”Hello World”;

$temp=$temp+10;

echo $temp;

//program terminates with no error.

* const x=1;

const y=”2”;

const z= x+y; //12

* value=21;

value=value+”dot”;

console.log(value);

4.Work on Data types? Write a Program for Datatypes in java?

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in the memory.

There are 2 types of Datatypes:

1.Primitive Datatyeps:

Primitive datatypes are predefined by the language and named by a keyword. Primitive values do not share state with other primitive values.

2. User-Defined Datatypes:

The non-primitive data types include [Classes](https://www.javatpoint.com/object-and-class-in-java), [Interfaces](https://www.javatpoint.com/interface-in-java), and [Arrays](https://www.javatpoint.com/array-in-java).

public class Datatypes {

public static void main(String[] args) {

int myint=1;

double mydouble = 2000.00;

float myfloat= 5.6666f;

char mychar= 'c';

String mystr="Virtusa";

boolean myboolean= false;

System.out.println(myint);

System.out.println(myfloat);

System.out.println(mychar);

System.out.println(mystr);

System.out.println(myboolean);

}

}

4.Write the Simplest code related to if-else,switch and looping structures?

If-else:

public class Ifelse {

public static void main(String[] args) {

int a= 100;

int b=200;

if (a>b)

{

System.out.println("a is highest no");

}

else

{

System.out.println("b is highest no");

}

}

}

Switch:

public class Switch {

public static void main(String[] args) {

int count=3;

switch(count)

{

case 0: System.out.println("Count is equal to 0");

break;

case 1: System.out.println("Count is equal to 1");

break;

case 2: System.out.println("Count is equal to 2");

break;

case 3: System.out.println("Count is equal to 3");

break;

default:System.out.println("Count is higher or lower than 4");

break;

}

}

}

For:

public class For {

public static void main(String[] args) {

for(int i=0;i<=50;i++) {

System.out.println("Value of i is"+i);

}

}

}

While:

public class While {

public static void main(String[] args) {

int j=1;

int i=15;

while(j<=i)

{

System.out.println(j);

j++;

}

}

}

Do-while:

public class Dowhile {

public static void main(String[] args){

int i=30;

do {

System.out.println(i);

i--;

}while(i>=10);

}

}

5. Create a class with a parameterised constructor use of data members and member functions?

class Library{

int Libid;

String Libname;

Library (int ID,String Name)

{

Libid=ID;

Libname=Name;

}

void library(){

System.out.println("Library id is:"+Libid);

System.out.println("Library name is:"+Libname);

}

}

class Paramconst {

public static void main(String[] args) {

Library lib=new Library(101,"Central Library");

lib.library();

}

}

6. Create a class having 2 constructors in a single class using both parameterised and non- parameterised?

public class Menu {

static String Menuorder;

static int Orderid;

public Menu() {

Menuorder="Icecream";

Orderid=111;

}

public Menu(String Menuorder,int Orderid)

{

this.Menuorder= Menuorder;

this.Orderid=Orderid;

}

public static void main(String[] args) {

Menu m=new Menu();

System.out.println(Menuorder);

System.out.println(Orderid);

}

}