# ***TASK3***

NAME: Alice mwaura

REG NO:20/02572

COURSE:DIT

TASK:TASK3

1.DIFFERENCE BETWEEN PRIMITIVE AND REFERENCE DATA TYPES

PRIMITIVE

Variables information is stored as the value of that variable examples of primitive datatypes are Int, float, double, Boolean and char.

Reference data types

Holds an address to information related to that variable.

2.DEFINE THE SCOPE OF A VARIABLE(HINT: local and global variable)

Scope of a variable means the lifetime of a variable in the program.

Local-variables that are declared inside a function or block.

Executes only the statements in that function

Global- Variables are described outside of all functions. Can be accessed by any function.

Method scope-variables are declared inside a method are available anywhere in the method following the line of code where they are declared.

Block scope-All codes between curly braces. Variables are declared inside blocks of codes are only accessible by the code between the curly braces which follows the line in which the variable was declared.

3.WHY IS INITIALIZATION OF VARIABLES REQUIRED.

Initialization of a variable is giving a variable a correct initial value.

1.The compiler cannot let you use uninitialized variable

2.2To avoid logical errors which are hard to debug.

3.If you don’t initialize your variable you will get a compilation error

4.uninittialized variable cannot be used until it is assigned.

4.DIFFERENTIATE BBETWEEN STATIC, LOCAL, AND INSTANCE VARIABLES.

STATIC-This variable is declared inside a class but outside of a method starting with a keyword static.

-they are accessible throughout the class

-static variable has the same lifetime as the program

LOCAL-They are declared within a method.

-the scope of the local variable is limited to the method it is declared inside.

-it starts its lifetime when the method is invoked.

INSTANCE-declared inside a class but outside of a method or block.

-accessible throughout the class

5, DIFFERENTIATE BETWEEN NARROW AND WIDENING CASTING

Casting-assigning a value of one primitive data type to another type.

1.widening

Converting a smaller type to a larger type

e.g. INT to DOUBLE

2.Narrow

Converting a larger type to a smaller size type

e.g. BYTE to INT

6.FILL THE FOLLOWING TABLE

|  |  |  |  |
| --- | --- | --- | --- |
| **TYPE** | **SIZE (IN BYTES)** | **DEFAULT** | **RANGE** |
| boolean | 1 bytes | false | true, false |
| Char | 2 bytes | \u0000’ | ‘\0000’ to ‘\ffff’ |
| Byte | 1byte | 0 | -27 to +27-1 |
| Short | 2bytes | 0 | -215 to +215-1 |
| Int | 4 bytes | 0 | -231 to +231-1 |
| Long | 8bytes | 0L | - 9.223….to9.223… |
| Float | 4 bytes | 00.0f | 3.40282347 |
| Double | 8 bytes | 0.0d | -1.8E+308 to +1.8E+308 |

7.IMPORTANCE OF JAVA PACKAGES

1.avoid naming conflicts.

2.To write a better maintainable code

3.implements data encapsulation

4.makes it easy to search or locate classes and interfaces

5.uniquely compare the classes in other packages

9.differentiate between containers and components as used in java

Containers –interface between a component and the low level platform. A component that can contain other components. It also a class or function that describes part of a UI.

Component –is the fundamental user interface object in java.

10. Write a Java program to reverse an array having five items of type int.

import java.util.;

public class Main

{

public static void main(String[] args) {

Integer[] intArray = {300,400,800,900,1000};

//this will print the array starting from the first number

System.out.println("Array before reversal:");

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

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

System.out.println();

//this wwill print the array starting from the last number

System.out.println("Main Array printed in reverse order:");

for(int i=intArray.length-1;i>=0;i--)

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

}

}

OUTPUT:

Array before reversal: 300 400 800 900 1000

Main Array printed in reverse order:1000 900 800 400 300

11.explain what is meant by event

Anything that can occur asynchronously not under the control of the program, to which the program might want to respond.

e.g. when a user clicks a mouse button, types a character, or clicks a button the program will respond to the click of the mouse button

12differntiate between

1.polymorphism and encapsulation

Polymorphism-allows program code to have different meaning or functions

Encapsulation- process of keeping classes private so they cannot be modified by external codes

2.method overloading and method overriding

Method overloading- When two or more methods in the same class have the same name but different parameters

Method over riding- when the method signature which is the name and parameters are the same in the super class and the child class.

3.Class and interface

Class- class can be inherited by another class using the keyword ‘extends’.

Has constructors.

Interface-can be inherited by a class using the keyword ‘implements’ and it can e inherited by another interface using the keyword ‘extends’

Has no constructors.

13 using examples, explain the two possible ways of implementing polymorphism. Show your code in java.

1Method loading

Is the process that can create multiple methods of the same name in the same class, and all the methods work in different ways, This occurs when there is more than one method of the same name in the class

class Pattern {

public void display() {

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

System.out.print("\*");

}

}

public void display(char symbol) {

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

System.out.print(symbol);

}

}

}

class Main {

public static void main(String[] args) {

Pattern d1 = new Pattern();

d1.display();

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

d1.display('#');

}

}

2. method overriding

The process when the subclass or a child has the same method as declared in the parent class.

class Language {

public void displayInfo() {

System.out.println("Common English Language");

}

}

class Java extends Language {

@Override

public void displayInfo() {

System.out.println("Java Programming Language");

}

}

class Main {

public static void main(String[] args) {

// create an object of Java class

Java j1 = new Java();

j1.displayInfo();

// create an object of Language class

Language l1 = new Language();

l1.displayInfo();

}

}

1. With relevant examples, explain the following concepts as used in Java programming.

a. Mutable classes.

Explain what is meant by mutable class

Objects whose value can be changed after initialization e.g java.util,date

Write a program that implements the concept of mutable class

package javaProgram;

public class Student

{

private String name;

private int regNo;

public Student(String name, int id)

{

this.name = name;

this.regNo = id;

}

public String getName() {

return name;

}

public int getRegNo() {

return regNo;

}

public void setName(String name) {

this.name = name;

}

public void setRegNo(int rollNo) {

this.regNo = rollNo;

}

}

package javaProgram;

public class Mutableexample{

public static void main(String[] args)

{

// Creating an object of mutable class.

Student st = new Student("Alice", 12);

String name = st.getName();

int regNo = st.getRegNo();

System.out.println("Name: " +name);

System.out.println("Regno: " +regNo);

st.setName("joy");

st.setRegNo(25);

String changeName = st.getName();

int changeRegNo = st.getRego();

System.out.println("Name after modification: " +changeName);

System.out.println("Reg no after modification: " +changeRegNo);

}

}

MY OUTPUT WIL BE:

Name:Alice

Reg no:12

Name after modification:joy

Reg no after modification:25

b. Immutable classes.

Explain what is meant by immutable class

Objects whose values cannot be changed after initialization.e.g int float

Write a program that implements the concept of immutable class

package javaProgram;

// An immutable class.

public final class Student

{

private final String name;

private final int regNo;

public Student(String name, int id)

{

this.name = name;

this.regNo = id;

}

public String getName() {

return name;

}

public int getRegNo() {

return regNo;

}

}

package javaProgram;

public class ImmutableDemo {

public static void main(String[] args)

{

Student st = new Student("Alice", 12);

String name = st.getName();

int rollNo = st.getReglNo();

System.out.println("Name: " +name);

System.out.println("Reg no: " +regNo);

}

}

OUTPUT:Alice

Reg no:12

2. Explain what a String buffer class is as used in Java

This is a peer class of string that provides much of the functionality of strings.It is a thread –safe mutable sequence of characters.

the syntax of creating an object of StringBuffer class

class StringBufferExample{

public static void main(String args[]){

StringBuffer sb=new StringBuffer("Hello ");

sb.append("Java");//now original string is changed

System.out.println(sb);//prints Hello Java

}

}

Explain the methods in the StringBuffer class

1.string buffer class operand () method

The append () method concanates the given argument with this string.

2.string buffer insert() method

The insert method inserts the given string with this string given at a given position.

3.string buffer replace method

The replace () method replaces the given string from the specified beginindex and endindex

4.string buffer delete() method

The delete method of the string buffer class deletes the string from specified begin index to end index

5string buffer reverse () method

Reverses the string builder class

b. Write the output of the following program.

This generate an error since there is no public class found to execute

';' expected after public static void main(String args[])

not a statement

String ast = "hello i love java"; this is not a statement and it can not be executed on the output

d. With explanation, write the output of the following program.

illegal start of type public static void main(String args[])

';' expected after public static void main(String args[])

not a statement StringBuffer bfobj = new StringBuffer("Jambo");

e. With explanation, write the output of the following program.

Error

';' expected after public static void main(String args[])

not a statement StringBuffer str1 = new StringBuffer("Jambo");

f. With explanation, write the output of the following program.

Error

illegal start of type class output