**21/02379 Karen Saru**

**DIT Java programming.**

**Assignment.**

**Task three.**

1. **Explain the difference between primitive and reference data types.**

Primitive data type are the basic type of data which store primitive values example byte, float, char, double while reference data type store addresses example string, scanner

1. **Define the scope of a variable(hint: local and global variable)**

Scope of a variable defines how a specific variable is accessible within the program or across classes

1. **Why is initialization of variables required?**

Local variables must be initialized before use, as they don’t have a default value and the compiler won’t let us use an uninitialized value.

1. **Differentiate between static, instance and local variables.**

Static variables are created when the program starts and destroyed when the program stops, while instance variables are created when an object is created with the use of the keyword ‘new’ and destroyed when the object is destroyed and finally local variable are variables created when a block method or constructor is started and the variable will be destroyed once it exists the block, method or constructor.

1. **Differentiate between widening and narrowing casting in java**

Widening casting automatically converts one data type to another data type while narrowing casting we manually convert one data type into another using the parenthesis.

1. **Explain the importance of using java packages**

Packages are useful to arrange related classes and interface into a group. This makes all the classes and interface performing the same task to put together in the same package.

Packages hide the classes and & interfaces in a separate subdirectory so that accidental deletion of classes & interfaces will not take place.

1. **Explain three controls used when creating GUI applications in java language**

Label is a component that is used to define a simple text on the screen.

The radio button is used to provide various options to the user. The user can only choose one option among all.

Text field is basically used to get the input from the user.

1. **Explain the difference between containers and components as used In java**

The class container is the superclass for the containers of AWT, the container object can contain other AWT components while the class component is the abstract base class for the non menu user interface controls of AWT

1. **Programs written for graphical user interface have to deal with “events”. Explain what is meant by the term event.**

The term event, it’s the change in the state of an object.

1. **Explain the difference between the following terms as used in java programming.**

**Polymorphism and encapsulation.**

Polymorphism allows program code to have different meaning or functions while encapsulation is the process of keeping classes private so they cannot be modified by external codes.

**Method overloading and method overriding.**

Method overloading is performed within class while method overriding occurs in two classes that have inheritance relationship.

**Class and interface.**

A class can inherit another class using extends keywords and implement an interface while interface can inherit only an interface.

**Inheritance and polymorphism.**

Inheritance supports the concept of reusability and reduces code length in object oriented programming while polymorphism allows the object to decide which form of the function to implement at overloading as well as overriding.

1. **Using examples, explain the two possible ways of implementing polymorphism. Show your code in java.**

* Compile time polymorphism

This type of polymorphism is achieved by function overloading. When there are multiple functions with the same name but different parameters then these functions are said to be overloaded.

* Runtime polymorphism.

A process in which function call to the overridden method is resolved at runtime. This type of polymorphism is achieved by method overriding.

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

A} **mutable classes. Explain what is meant by mutable class.**

* a mutable class is one that can change its internal state after it is created.

B} **immutable classes. Explain what is meant by immutable class.**

* Immutable classes define objects which once created never change their value

**C} explain the situations where mutable classes are more preferable than immutable classes when writing a java program.**

When you have large objects, creating a single copy becomes expensive. So, creating multiple copies of an object becomes less efficient and slows down your computer’s operation. In such a situation, a mutable object is better than an immutable object.

1. **Explain what a string buffer class is as used in java**

String buffer is a peer class of string that provides much of the functionality of strings.

**Syntax of creating an object of string buffer class.**

public final class StringBuffer

extends Object

implements Serializable, CharSequence

**Explain the methods in the string buffer class.**

* Insert () method. The insert () method inserts the given string with this string at the given position
* Replace () method. The replace () method replaces the given string from the specified begin index and end index -1
* Delete () method. The delete () method of a string buffer class deletes the string from the specified begin index to endindex-1
* Reverse () method. The reverse method of string builder class reverses the current string.
* Capacity () method. The capacity method of string buffer class returns the current capacity of the buffer

**D} write the output of the program.**

The output of the program will be Jambo.

The append (bfobj1);

Since the append is a string builder disposed the temporary garbage in this case it was Kenya.

E} **with explanation write the output of the program.**

The output of the program will be Jambo.

1. The following table shows data type, its size, default value and the range. Filling in the missing values.

|  |  |  |  |
| --- | --- | --- | --- |
| **TYPE** | **SIZE (IN BYTES)** | **DEFAULT** | **RANGE** |
| boolean | 1 bit | False | true, false |
| Char | 2 bytes | ‘\u0000’ | ‘\0000’ to ‘\ffff’ |
| Byte | 1 byte | 0 | -27 to +27-1 |
| Short | 2 bytes | 0 | -215 to +215-1 |
| Int | 4 bytes | 0 | -231 to +231-1 |
| Long | 8 bytes | 0L | - |
| Float | 4 bytes | 00.0f | Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits |
| Double | 8 bytes | 0.0d | -1.8E+308 to +1.8E+308 |