# **Task Three Assignment**

1. Explain the differences between primitive and reference data types.
   1. **Primitive variables take less memory than reference variables because they don't need to maintain object metadata e.g. object header.**
   2. **Primitive variables are created in the stack while reference variable is created in heap space, which is managed by Garbage Collector.**
   3. **When you return primitive types from a method then the primitive value is returned but when you return a reference type, again only handle to the is returned**
   4. **When you pass primitive values to a method the values are passed to the method, but when you pass the reference variable, only the handle is copied. Which means for primitives, changing the formal parameter's value doesn't affect the actual parameter's value.**
   5. **When you compare primitive variables using equality (==) operator, their primitive values are compared but when you compare reference variable, their address is compared, which means two objects which are logically equal.**
   6. **When you assign a value to primitive data types, the primitive value is copied, but when you assign an object to reference type, the handle is copied. Which means for reference type object is not copied only the handle is copied.**
   7. **Primitive types stores values but reference type stores handle objects in heap space.**
2. Define the scope of a variable (hint: local and global variable)

**defines how a specific variable is accessible within the program or across classes.**

**Local variables are the variables that are declared inside a method, constructor, or block have a method-level or block-level scope and cannot be accessed outside in which it is defined**

1. Why is initialization of variables required.

**We initialize a variable so as to give it a correct initial value**. **When a variable is declared, it will point to a piece of memory. Accessing the value of the variable will give you the contents of that piece of memory. However until the variable is initialised, that piece of memory could contain anything**

1. Differentiate between static, instance and local variables

**Static variables are declared with the static keyword in a class, but outside a method, constructor or a block**

**Instance variable are declared in a class but outside a method, constructor or any block.**

**Local variable are variables within programming blocks or subroutines.**

1. Differentiate between widening and narrowing casting in java.

**Widening Casting (automatically)  involves the conversion of a smaller data type to the larger type size while Narrowing Casting (manually) involves converting a larger data type to a smaller size type.**

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 | “\u0000’ | ‘\0000’ to ‘\ffff’ |
| Byte | 1 | 0 | -27 to +27-1 |
| Short | 2 | 0 | -215 to +215-1 |
| Int | 4 | 0 | -231 to +231-1 |
| Long | 8 | 0L | -[-263, 263-1] |
| Float | 4 | 00.0f | -3.4E+38to+3.4E+38 |
| Double | 8 | 0.0d | -1.8E+308 to +1.8E+308 |

1. Explain the importance of using Java packages

**Java package is used to categorize the classes and interfaces so that they can be easily maintained**

1. Explain three controls used when creating GUI applications in Java language.
   1. **Textbox – enables to edit or enter text**
   2. **Checkbox - enables to select or deselect an option**
   3. **Button – it’s a widget which is clickable and provide user a simple way to trigger an event**
2. Explain the difference between containers and components as used in Java.

**A container is a java class instantiated to hold groups of components e.g JFrame while components is a java class instantiated to create a type of GUI object.**

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

(**GIT HUB as question 10. Reverse arrays)**

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

**This is the change in the state of an object or behavior by performing actions**

1. Give at least two different examples of events, and discuss how a program might respond to those events.
   1. **Foreground event - events that require user interaction to generate with GUI components i.e. if user clicks a button an event will be fired.**
   2. **Background event – don’t require any user interaction. They automatically generate in the background**
2. Explain the difference between the following terms as used in Java programming.
   1. Polymorphism and encapsulation

**Polymorphism is the ability to process objects differently depending on their data types while encapsulation is the process of keeping classes private so they cannot be modified by external codes**

* 1. method overloading and method overriding

**Method overloading is a feature that allows a class to have more than one method having the same name but their parameters are different while method overriding allows a subclass to implement a method that is already declared by one of its super classes**

* 1. class and interface

**A class is a blueprint from which individual objects are created while an interface is a reference type in java and is a collection of abstract methods**

* 1. inheritance and polymorphism

**in inheritance it supports the concept of code reusability and reduces the length of the code in the program while polymorphism allows the object to decide which form of the function to implement at compile time and run time**

1. Using examples, explain the two possible ways of implementing polymorphism. Show your code in java.
   * + 1. **Method Overriding – its used to provide the specific implementation of a method which is already provided by its superclass. Its for Run-time Polymorphism.**
       2. **Method overloading – it increases the readability of the program**

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

a. Mutable classes.

1. Explain what is meant by mutable class

**It’s an object which you can change the fields and state after the object is created e.g. java.util.**

1. Write a program that implements the concept of mutable class

**(GIT HUB as question 14. Mutable classes)**

b. Immutable classes.

* 1. Explain what is meant by immutable class

**This are objects which cannot change anything once the object is created e.g. integer**

* 1. Write a program that implements the concept of immutable class

**(GIT HUB as question 14. Immutable classes)**

c. Explain the situations where mutable classes are more preferable than immutable classes when writing a Java program.

1. **In mutable classes the fields can be changed after object creation while in immutable classes fields can’t be changed once created.**
2. **Mutable classes provide methods to modify fields value while immutable classes don’t have any method to modify field value**
3. **Mutable classes have getter and setter methods while immutable classes only have getter methods**

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

**It’s used to create mutable string. It’s the same thing as string class except its mutable i.e. it can be changed.**

* 1. the syntax of creating an object of StringBuffer class

**StringBuffer sb = new StringBuffer();**

* 1. Explain the methods in the StringBuffer class
  2. **Append () – its used to concatenates the given argument with the string.**
  3. **Insert () – inserts the given string with the string at the given position.**
  4. **Replace () – replaces the given string from the specified beginIndex and EndIndex.**
  5. **Delete() – deletes string specified beginIndex to EndIndex.**
  6. **reverse ()– it reverse the current string**
  7. **substring(int beginIndex, int endIndex) – It is used to return the substring from the specified beginIndex and endIndex**

b. Write the output of the following program.

class Myoutput

1. {
2. public static void main(String args[])
3. {
4. String ast = "hello i love java";
5. System.out.println(ast.indexOf('e')+" "+ast.indexOf('ast')+" "+ast.lastIndexOf('l')+" "+ast .lastIndexOf('v'));
6. }
7. }

c. Explain your answer in (2b) above.

**Command execution failed since there was an expected ‘;’ that wasn’t on the program**

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

**Command execution fail since c is not recognized**

class Myoutput

1. {
2. public static void main(String args[])
3. {
4. StringBuffer bfobj = new StringBuffer("Jambo");
5. StringBuffer bfobj1 = new StringBuffer(" Kenya");
6. c.append(bfobj1);
7. System.out.println(bfobj);
8. }
9. }

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

**Output – obmaJ**

**The program is created to reverse the string entered**

class Myoutput

1. {
2. public static void main(String args[])
3. {
4. StringBuffer str1 = new StringBuffer("Jambo");
5. StringBuffer str2 = str1.reverse();
6. System.out.println(str2);
7. }
8. }

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

**Output - 1 is a digit**

**- a is a lower case letter**

**The program allows user to distinguish the character given in**

**class Myoutput**

1. {
2. class output
3. {
4. public static void main(String args[])
5. {
6. char c[]={'A', '1', 'b' ,' ' ,'a' , '0'};
7. for (int i = 0; i < 5; ++i)
8. {
9. i++;
10. if(Character.isDigit(c[i]))
11. System.out.println(c[i]+" is a digit");
12. if(Character.isWhitespace(c[i]))
13. System.out.println(c[i]+" is a Whitespace character");
14. if(Character.isUpperCase(c[i]))
15. System.out.println(c[i]+" is an Upper case Letter");
16. if(Character.isLowerCase(c[i]))
17. System.out.println(c[i]+" is a lower case Letter");
18. i++;
19. }
20. }
21. }