Course 3 – Sprint 2 – Objects and Constructors – Assessment 2

Fix the errors

# Predict the output of the below code

String str1 = "Fruit"; String str2 = "Fruit";

boolean result = str1.equals(str2); System.out.println(result);

true

# Predict the output of the below code snippet

String str1 = "fruits";

String str2 = "fruits are good";

int result = str1.compareTo(str2); System.out.println(result);

-9

# Predict the output of the below code snippet

int a = 10; double b = 2.00;

char[] arr = {'a', 'b'};

System.out.println("Return Value : " + String.valueOf(a) ); System.out.println("Return Value : " + String.valueOf(b) );

Return Value : 10

Return Value : 2.0

# Will the below piece of code compile? Give reasons for your answer

class Person {

final long MAX\_AGE; Person(){

MAX\_AGE = 90;

}

}

Yes the code will compile as we are initializing the variable MAX\_AGE only once.

1. You run a bookstore that sells all types of books on fiction, religion, adventure, cookbooks etc. written by different authors. The end of year sale is on and you wish to give a discount on books belonging to a certain genre. All fiction books are given a 25% discount.

# Identify the objects and behaviour in the above problem statement. Declare the attributes of the Objects Identified and use appropriate datatypes

**Attributes**

* String title
* String author
* String genre
* float price

**Behavior**

* public float getDiscountedPrice()

{

float discountedPrice = price;

if(genre.equals(“Fiction”))

{

discountedPrice = price \* 0.75;

}

return discountedPrice;

}

1. **Find the error in the below code and rectify it for successful execution of the program.**

class Student{

public static int calculateTotal(int mark1,int mark2,int mark3)

{

return mark1+mark2+mark3;

}

public static void main(String[] args) {

static int total = Student.*calculateTotal*(88,99,80); System.*out*.println("The total marks : "+total);

}

}

class Student

{

public static int calculateTotal(int mark1,int mark2,int mark3)

{

return mark1+mark2+mark3;

}

public static void main(String[] args)

{

int total = Student.*calculateTotal*(88,99,80); System.*out*.println("The total marks : "+total); *// no static modifier for variable*

}

}

# Write a no-argument constructor for the below Person class.

public class Person{ int age;

String name;

String address;

String phoneNumber; String email;

}

public class Person

{

int age;

String name;

String address;

String phoneNumber;

String email;

public Person()

{

age = -1;

name = "";

address = "";

phoneNumber = "";

email = "";

}

}

# Write a parameterized constructor for the class given below, also initialize an object of the class using the new keyword.

class Customer{ String name;

int age;

String [] addresses; long mobileNumber;

boolean isLoyaltyCustomer;

}

public class Customer

{

String name;

int age;

String[] addresses;

long mobileNumber;

boolean isLoyaltyCustomer;

public Customer(String name, int age, String[] addresses, long mobileNumber, boolean isLoyaltyCustomer)

{

this.name = name;

this.age = age;

this.addresses = addresses;

this.mobileNumber = mobileNumber;

this.isLoyaltyCustomer = isLoyaltyCustomer;

}

}

# Predict the output of the following code: \*

public static void main( String[] args )

{

String string1 = "Hello World"; System.*out*.println(string1.substring(1,5));

}

ello

# Predict the output for the following code \*

public static void main(String[] args) { String str1 = "Java";

String str2 = "Programming"; String str3 = str1.concat(str2); System.*out*.println(str3);

}

JavaProgramming

# Which of the following statements are false for Local Variables: \*

* 1. Objects do not maintain copies of local variables. 2.Not Mandatory to provide initial values.

1. The variables are declared within a method but do not get any default values and must be initialized with a value before using them.
2. The scope is generally limited to a method and starts from the line they are declared. The scope ends within the block of code

2.

# Which of the following statements are true for constructor in java? \*

* 1. The constructor must have the same name as the class name. 2.The constructors **are called only once** during object initialization. 3.They cannot be called explicitly like methods using the dot operator.

4.The variables declared in the class cannot be given any initial values inside the constructor.

1, 2 and 3

# The following code is giving compile time error fix the compile time error \*

**in the given code:**

public class Employee

{

int salary; String name; int empId;

public void display(){ String designation;

System.*out*.println("salary is : " + salary); System.*out*.println("Name is : " + name); System.*out*.println("Emp Id is : " + empId); System.*out*.println("Designation is : " + designation);

}

public static void main(String[] args) { Employee employee = new Employee(); employee.display();

}

}

public class Employee

{

int salary;

String name;

int empId;

String designation;

public void display()

{

System.out.println("salary is : " + salary);

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

System.out.println("Emp Id is : " + empId);

System.out.println("Designation is : " + designation);

}

public static void main(String[] args)

{

Employee employee = new Employee();

employee.display();

}

}

15. \*

public class Employee

{

int salary;

String name;

String designation; int empId;

Employee(){

}

public Employee(int salary, String name, String designation, int empId) { salary = salary;

name = name; designation = designation; empId = empId;

}

public void display(){ System.*out*.println("Name is : " + name); System.*out*.println("Salary is : " + salary); System.*out*.println("EmpId is : " + empId);

System.*out*.println("Designation is : " + designation);

}

public static void main(String[] args) {

Employee employee = new Employee(350000,"John","XYZ",1001); employee.display();

}

}

The above code is giving following output.

# Name is : null Salary is : 0 EmpId is : 0

**Designation is : null**

Fix the issue in the code to get following output

# Name is : John

**Salary is : 350000 EmpId is : 1001**

# Designation is : XYZ

public class Employee

{

int salary;

String name;

String designation;

int empId;

Employee()

{

}

// this keyword missing in the ctor.

public Employee(int salary, String name, String designation, int empId)

{

this.salary = salary;

this.name = name;

this.designation = designation;

this.empId = empId;

}

public void display()

{

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

System.out.println("Salary is : " + salary);

System.out.println("EmpId is : " + empId);

System.out.println("Designation is : " + designation);

}

public static void main(String[] args)

{

Employee employee = new Employee(350000,"John","XYZ",1001);

employee.display();

}

}

16. **Is this a correct definition for the main method ?**

void static main(String args){

}

No this is incorrect. It has to be ‘**public static void** main(String**[]** args)’

public is necessary.

Order of static and void needs to be correct.

main can only accept array of strings.