Subject : Advanced Java Programming Prepared by: Mrs. Nagamma Aravalli

Subject Code: 21CS642 Assistant Prof., Dept. Of CSE,AIT

Date: 15-05-2024

**Module -1: Enumerations,Autoboxing and Annotations**

**Enumeration Practice Programs With Solutions**

1. Write a Java program to create an enum called ‘WeekDays’ representing the days of the week.



| enum WeekDays{  SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY } public class EnumDemo1 {  public static void main(String[] args) {  WeekDays today;  today= WeekDays.WEDNESDAY;  //output of an enum value  System.out.println("Today is : "+today);  } } |
| --- |

**Output:**

Today is : WEDNESDAY

1. Write a Java program to check equality of two constants of the same enumeration using relational operator.

| enum WeekDays{  SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY } public class EnumDemo2 {  public static void main(String[] args) {  WeekDays today;  today= WeekDays.WEDNESDAY;  //output of an enum value  System.out.println("Today is : "+today);  if(today == WeekDays.WEDNESDAY)   {  System.out.println("today variable contains WEDNESDAY");  }  } } |
| --- |

**Output:**

Today is : WEDNESDAY

today variable contains WEDNESDAY

1. Write a Java program to demonstrate enum constant using Switch statement

| enum WeekDays //Enumeration defined {  SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY  }  class EnumDemo3 {  public static void main(String args[])  {  WeekDays wk = WeekDays.SUNDAY;    switch(day)  {  case SUNDAY:   System.out.println("sunday is fun day");  break;  case MONDAY:   System.out.println("monday");  break;  default:  System.out.println("other day");  }  }  } |
| --- |

**Output:**

Sunday is fun day

1. Write a Java program to demonstrate a loop through the enum.

| // use of enumeration built-in method values() enum Apple {  Jonathan, GoldenDel, RedDel, Winesap, Cortland } class EnumDemo4  {  public static void main(String args[])  {   Apple app;  System.out.println("Here are all Apple constants:");  // use values()  Apple allapples[] = Apple.values();  for(Apple a : allapples)  System.out.println(a);  }  } |
| --- |

**Output:**

Here are all Apple constants:

Jonathan

GoldenDel

RedDel

Winesap

Cortland

1. Write a Java Program to demonstrate to get enum constant of specified string value.

| // use of enumeration built-in method valueOf() import java.util.\*; enum Apple {  Jonathan, GoldenDel, RedDel, Winesap, Cortland } public class EnumDemo5 {  public static void main(String args[])  {   // Get the variety name of the apple from the user  System.out.println("Enter an Apply name: ");  Scanner scanner = new Scanner(System.in);  String appleString = scanner.nextLine();    // Use the valueOf() method to convert the string to an enum value  Apple apple = Apple.valueOf(appleString);  // Print the Variety of an Apple  System.out.println("The Variety of an apple is: " + apple);  } } |
| --- |

**OutPut:**

Enter an Apply name:

RedDel

The Variety of an apple is: RedDel

1. Write a Java program to demonstrate default constructor of an enumeration

| enum WeekDays{  SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY;  WeekDays(){  System.out.println("Default constructor called: "+this);  } } public class EnumDefaultConstructor {  public static void main(String args[])  {  WeekDays day;  // call to default constructor of an constant MONDAY   day = WeekDays.MONDAY;   }   } |
| --- |

**OutPut:**

Default constructor called: SUNDAY

Default constructor called: MONDAY

Default constructor called: TUESDAY

Default constructor called: WEDNESDAY

Default constructor called: THURSDAY

Default constructor called: FRIDAY

Default constructor called: SATURDAY

1. Write a Java program for the following problem statement:

Create an enum called Day with the following constants: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, and SUNDAY. Each constant should have a private instance variable called number that represents the day's number in the week (e.g., MONDAY should have a number of 1, TUESDAY should have a number of 2, and so on). The enum should also have a constructor that takes an int value and assigns it to the number instance variable. Finally, the enum should have a method called getDayNumber() that returns the value of the number instance variable.

| enum Day {  MONDAY(1), TUESDAY(2), WEDNESDAY(3), THURSDAY(4), FRIDAY(5), SATURDAY(6), SUNDAY(7);  private int number;  Day(int number) {  this.number = number;  }  public int getDayNumber() {  return number;  } }  public class EnumAsClass {  public static void main(String[] args) {  Day day = Day.MONDAY;  System.out.println(day +" its number is: "+day.getDayNumber()); // 1  } } |
| --- |

**Output:**

MONDAY its number is: 1

1. Write a Java Program to print position of all the constants of an enum

| enum Apple { Jonathan, GoldenDel, RedDel, Winesap, Cortland } class EnumDemo8 {  public static void main(String args[])  {  // Obtain all ordinal values using ordinal().  System.out.println("Here are all apple constants" +  " and their ordinal values: ");  for(Apple a : Apple.values())  System.out.println(a + " " + a.ordinal());  } } |
| --- |

**Output**:

Here are all apple constants and their ordinal values:

Jonathan 0

GoldenDel 1

RedDel 2

Winesap 3

Cortland 4

1. Write a program to compare positions of two enum constants of same enumeration

| **enum Apple4 { Jonathan, GoldenDel, RedDel, Winesap, Cortland } public class EnumDemo9 {  public static void main(String args[])  {  Apple4 ap1, ap2, ap3;  ap1 = Apple4.RedDel;  ap2 = Apple4.GoldenDel;  ap3 = Apple4.RedDel;  System.out.println();  // Demonstrate compareTo()  if(ap1.compareTo(ap2) < 0)  System.out.println(ap1 + " comes before " + ap2);  if(ap1.compareTo(ap2) > 0)  System.out.println(ap2 + " comes before " + ap1);  if(ap1.compareTo(ap3) == 0)  System.out.println(ap1 + " equals " + ap3);   } }** |
| --- |

**Output:**

GoldenDel comes before RedDel

RedDel equals RedDel

1. Write a program to check equality of two eum constants.

| **enum Apple6{ Jonathan, GoldenDel, RedDel, Winesap, Cortland } enum TestApple{  Jonathan, GoldenDel, RedDel, Winesap, Cortland } public class EnumDemo10 {**  **public static void main(String args[])  {  Apple6 ap1, ap2, ap3;  TestApple t1;**  **ap1 = Apple6.Cortland;  ap2 = Apple6.Jonathan;  t1 = TestApple.Jonathan;  ap3 = Apple6.Cortland;    System.out.println(ap1+" and "+ ap2+" from Apple6 are equal: "+ap1.equals(ap2));  System.out.println(ap2+" and "+ t1+" from Apple6 and TestApple are equal: "+ap2.equals(t1));  System.out.println(ap1+" and "+ ap3+" from Apple6 are equal: "+ap1.equals(ap3));  } }** |
| --- |

**Output:**

Cortland and Jonathan from Apple6 are equal: false

Jonathan and Jonathan from Apple6 and TestApple are equal: false

Cortland and Cortland from Apple6 are equal: true