**package** com.assignment;

**public** **class** Question\_No\_1 {

/\*Declare a method-local variable status of type boolean with the value true and convert it to a String using the toString method.

(Hint: Use Boolean.toString(Boolean) ).\*/

**public** **static** **void** m1() {

**boolean** status=**true**;

String str = Boolean.*toString*(**true**);

System.***out***.println(str);

}

/\*Declare a method-local variable strStatus of type String with the value "true" and convert it to a boolean using the parseBoolean method.

(Hint: Use Boolean.parseBoolean(String)).\*/

**public** **static** **void** m2() {

String strStatus="true";

**boolean** str = Boolean.*parseBoolean*(strStatus);

System.***out***.println(str);

}

/\*Declare a method-local variable strStatus of type String with the value "1" or "0" and attempt to convert it to a boolean.

(Hint: parseBoolean method will not work as expected with "1" or "0").

\*/

**public** **static** **void** m3() {

String strStatus="0";

String strStatus1="1";

**boolean** parseBoolean = Boolean.*parseBoolean*(strStatus);

**boolean** parseBoolean1 = Boolean.*parseBoolean*(strStatus1);

System.***out***.println(parseBoolean);

System.***out***.println(parseBoolean1);

}

/\*Declare a method-local variable status of type boolean with the value true and convert it to the corresponding wrapper class using Boolean.valueOf().

(Hint: Use Boolean.valueOf(boolean)).\*/

**public** **static** **void** m4() {

Boolean b=**true**;

Boolean valueOf = Boolean.*valueOf*(**true**);

System.***out***.println(valueOf);

}

/\*Declare a method-local variable strStatus of type String with the value "true" and convert it to the corresponding wrapper class using Boolean.valueOf().

(Hint: Use Boolean.valueOf(String)).\*/

**public** **static** **void** m5() {

String strStatus="true";

Boolean valueOf = Boolean.*valueOf*(strStatus);

System.***out***.println(valueOf);

}

/\*Experiment with converting a boolean value into other primitive types or

vice versa and observe the results\*/

**public** **static** **void** convertingBooleanValue() {

**boolean** x=**true**;

**boolean** y=**false**;

String str="Rushikesh";

**int** compare = Boolean.*compare*(x, y);

System.***out***.println(compare);

**boolean** parseBoolean = Boolean.*parseBoolean*("true");

System.***out***.println(parseBoolean);

String string = Boolean.*toString*(**true**);

System.***out***.println(string);

Boolean valueOf = Boolean.*valueOf*(str);

System.***out***.println(valueOf);

**int** hashCode = Boolean.*hashCode*(x);

System.***out***.println(hashCode);

}

**public** **static** **void** main(String[] args) {

//m1();

//m2();

//m3();

//m4();

//m5();

//convertingBooleanValue();

}

}

**package** com.assignment;

**public** **class** Question\_No\_2 {

/\*

\* Write a program to test how many bytes are used to represent a byte value

\* using the BYTES field. (Hint: Use Byte.BYTES).

\*/

/\*

\* Write a program to find the minimum and maximum values of byte using the

\* MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Byte.MIN\_VALUE and

\* Byte.MAX\_VALUE).

\*/

**public** **static** **void** m1() {

**byte** b = 12;

**int** bytes = Byte.***BYTES***;

System.***out***.println(bytes);

**byte** maxValue = Byte.***MAX\_VALUE***;

System.***out***.println(maxValue);

**byte** minValue = Byte.***MIN\_VALUE***;

System.***out***.println(minValue);

}

/\*

\* Declare a method-local variable number of type byte with some value and

\* convert it to a String using the toString method. (Hint: Use

\* Byte.toString(byte)).

\*/

**public** **static** **void** m2() {

**byte** b=24;

String string = Byte.*toString*(b);

System.***out***.println(string);

}

/\* Declare a method-local variable strNumber of type String with some value and convert it to a byte value using the parseByte method.

(Hint: Use Byte.parseByte(String)).\*/

**public** **static** **void** m3() {

String strNumber="12";

**byte** parseByte = Byte.*parseByte*(strNumber);

System.***out***.println(parseByte);

}

/\*Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a byte value.

(Hint: parseByte method will throw a NumberFormatException).\*/

**public** **static** **void** m4() {

String strNumber="Ab12Cd3";

**byte** parseByte = Byte.*parseByte*(strNumber);

System.***out***.println(parseByte);

}

/\* Declare a method-local variable number of type byte with some value and convert it to the corresponding wrapper class using Byte.valueOf().

(Hint: Use Byte.valueOf(byte)).\*/

**public** **static** **void** m5() {

**byte** b=15;

Byte valueOf = Byte.*valueOf*(b);

System.***out***.println(valueOf);

}

/\* .Declare a method-local variable strNumber of type String with some byte value and convert it to the corresponding wrapper class using Byte.valueOf().

(Hint: Use Byte.valueOf(String)).\*/

**public** **static** **void** m6() {

String strNumber="12";

Byte valueOf = Byte.*valueOf*(strNumber);

System.***out***.println(valueOf);

}

/\*Experiment with converting a byte value into other primitive types or

vice versa and observe the results.\*/

**public** **static** **void** m7() {

**byte** b=14;

**int** size = Byte.***SIZE***;

System.***out***.println(size);

Class<Byte> type = Byte.***TYPE***;

System.***out***.println(type);

**int** hashCode = Byte.*hashCode*(b);

System.***out***.println(hashCode);

**long** unsignedLong = Byte.*toUnsignedLong*(b);

System.***out***.println(unsignedLong);

}

**public** **static** **void** main(String[] args) {

//m1();

//m2();

//m3();

//m4();

//m5();

//m6();

//m7();

}

}

**package** com.assignment;

**public** **class** Question\_No\_3 {

/\*Write a program to test how many bytes are used to represent a short value using the BYTES field.

(Hint: Use Short.BYTES).\*/

**public** **static** **void** m1() {

**short** s=10;

**int** bytes = Short.***BYTES***;

System.***out***.println(bytes);

}

/\*Write a program to find the minimum and maximum values of short using the MIN\_VALUE and MAX\_VALUE fields.

(Hint: Use Short.MIN\_VALUE and Short.MAX\_VALUE).\*/

**public** **static** **void** m2() {

**short** s=13;

**short** maxValue = Short.***MAX\_VALUE***;

System.***out***.println("Max Value : "+maxValue);

**short** minValue = Short.***MIN\_VALUE***;

System.***out***.println("Min Value : "+minValue);

}

/\* Declare a method-local variable number of type short with some value and convert it to a String using the toString method.

(Hint: Use Short.toString(short)).\*/

**public** **static** **void** m3() {

**short** s=9;

String string = Short.*toString*(s);

System.***out***.println(s);

}

/\*Declare a method-local variable strNumber of type String with some value and convert it to a short value using the parseShort method.

(Hint: Use Short.parseShort(String)).\*/

**public** **static** **void** m4() {

String strNumber="12";

**short** parseShort = Short.*parseShort*(strNumber);

System.***out***.println(parseShort);

}

/\*Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a short value.

(Hint: parseShort method will throw a NumberFormatException).\*/

**public** **static** **void** m5() {

String strNumber="Ab12Cd3";

**short** parseShort = Short.*parseShort*(strNumber);

System.***out***.println(parseShort);

}

/\* Declare a method-local variable number of type short with some value and convert it to the corresponding wrapper class using Short.valueOf().

(Hint: Use Short.valueOf(short)).\*/

**public** **static** **void** m6() {

**short** s=16;

Short valueOf = Short.*valueOf*(s);

System.***out***.println(valueOf);

}

/\* Declare a method-local variable strNumber of type String with some short value and convert it to the corresponding wrapper class using Short.valueOf().

(Hint: Use Short.valueOf(String)).\*/

**public** **static** **void** m7() {

String strNumber="11";

Short valueOf = Short.*valueOf*(strNumber);

System.***out***.println(valueOf);

}

**public** **static** **void** main(String[] args) {

//m1();

//m2();

//m3();

//m4();

//m5();

//m6();

//m7();

}

}

**package** com.assignment;

**public** **class** Question\_No\_4 {

/\* Write a program to test how many bytes are used to represent an int value using the BYTES field.

(Hint: Use Integer.BYTES).\*/

**public** **static** **void** m1() {

**int** i=10;

**int** bytes = Integer.***BYTES***;

System.***out***.println(bytes);

}

/\* Write a program to find the minimum and maximum values of int using the MIN\_VALUE and MAX\_VALUE fields.

(Hint: Use Integer.MIN\_VALUE and Integer.MAX\_VALUE).\*/

**public** **static** **void** m2() {

**int** i=15;

**int** maxValue = Integer.***MAX\_VALUE***;

System.***out***.println(maxValue);

**int** minValue = Integer.***MIN\_VALUE***;

System.***out***.println(minValue);

}

/\*Declare a method-local variable number of type int with some value and convert it to a String using the toString method.

(Hint: Use Integer.toString(int)).\*/

**public** **static** **void** m3() {

**int** i=14;

String string = Integer.*toString*(i);

System.***out***.println(string);

}

/\* Declare a method-local variable strNumber of type String with some value and convert it to an int value using the parseInt method.

(Hint: Use Integer.parseInt(String)).\*/

**public** **static** **void** m4() {

String strNumber="100";

**int** parseInt = Integer.*parseInt*(strNumber);

System.***out***.println(parseInt);

}

/\*Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to an int value.

(Hint: parseInt method will throw a NumberFormatException).\*/

**public** **static** **void** m5() {

String strNumber="Ab12Cd3";

**int** parseInt = Integer.*parseInt*(strNumber);

System.***out***.println(parseInt);

}

/\*Declare a method-local variable number of type int with some value and convert it to the corresponding wrapper class using Integer.valueOf().

(Hint: Use Integer.valueOf(int)).\*/

**public** **static** **void** m6() {

**int** i=105;

Integer valueOf = Integer.*valueOf*(i);

System.***out***.println(valueOf);

}

/\* Declare a method-local variable strNumber of type String with some integer value and convert it to the corresponding wrapper class using Integer.valueOf().

(Hint: Use Integer.valueOf(String)).\*/

**public** **static** **void** m7() {

String strNumber="12";

Integer valueOf = Integer.*valueOf*(strNumber);

System.***out***.println(valueOf);

}

/\*Declare two integer variables with values 10 and 20, and add them using a method from the Integer class.

(Hint: Use Integer.sum(int, int)).\*/

**public** **static** **void** addition() {

**int** a=10;

**int** b=20;

**int** sum = Integer.*sum*(a, b);

System.***out***.println(sum);

}

/\*Declare two integer variables with values 10 and 20, and find the minimum and maximum values using the Integer class.

(Hint: Use Integer.min(int, int) and Integer.max(int, int)).\*/

**public** **static** **void** min\_max\_value() {

**int** a=10;

**int** b=20;

**int** min = Integer.*min*(a, b);

System.***out***.println("Minimum Value : "+min);

**int** max = Integer.*max*(a, b);

System.***out***.println("Maximum Value : "+max);

}

/\*Declare an integer variable with the value 7.

Convert it to binary, octal, and hexadecimal strings using methods from the Integer class.

(Hint: Use Integer.toBinaryString(int), Integer.toOctalString(int), and Integer.toHexString(int)).\*/

**public** **static** **void** m9() {

**int** i=7;

String binaryString = Integer.*toBinaryString*(i);

System.***out***.println("Binary : "+binaryString);

String octalString = Integer.*toOctalString*(i);

System.***out***.println("Octal : "+octalString);

String hexString = Integer.*toHexString*(i);

System.***out***.println("Hexadecimal : "+hexString);

}

**public** **static** **void** main(String[] args) {

//m1();

//m2();

//m3();

//m4();

//m5();

//m6();

//m7();

//addition();

//min\_max\_value();

//m9();

}

}

**package** com.assignment;

**public** **class** Question\_No\_5 {

/\* Write a program to test how many bytes are used to represent a long value using the BYTES field.

(Hint: Use Long.BYTES).\*/

**public** **static** **void** m1() {

**long** l=12345;

**int** bytes = Long.***BYTES***;

System.***out***.println(bytes);

}

/\*Write a program to find the minimum and maximum values of long using the MIN\_VALUE and MAX\_VALUE fields.

(Hint: Use Long.MIN\_VALUE and Long.MAX\_VALUE).\*/

**public** **static** **void** m2() {

**long** l=12345;

**long** maxValue = Long.***MAX\_VALUE***;

System.***out***.println(maxValue);

**long** minValue = Long.***MIN\_VALUE***;

System.***out***.println(minValue);

}

/\* Declare a method-local variable number of type long with some value and convert it to a String using the toString method.

(Hint: Use Long.toString(long)).\*/

**public** **static** **void** m3() {

**long** a=12345;

String string = Long.*toString*(a);

System.***out***.println(string);

}

/\*Declare a method-local variable strNumber of type String with some value and convert it to a long value using the parseLong method.

(Hint: Use Long.parseLong(String)).\*/

**public** **static** **void** m4() {

String strNumber="12345";

**long** parseLong = Long.*parseLong*(strNumber);

System.***out***.println(parseLong);

}

/\* Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a long value.

(Hint: parseLong method will throw a NumberFormatException).\*/

**public** **static** **void** m5() {

String strNumber="Ab12Cd3";

**long** parseLong = Long.*parseLong*(strNumber);

System.***out***.println(parseLong);

}

/\* Declare a method-local variable number of type long with some value and convert it to the corresponding wrapper class using Long.valueOf().

(Hint: Use Long.valueOf(long)).\*/

**public** **static** **void** m6() {

**long** a=12345;

Long valueOf = Long.*valueOf*(a);

System.***out***.println(valueOf);

}

/\*Declare a method-local variable strNumber of type String with some long value and convert it to the corresponding wrapper class using Long.valueOf().

(Hint: Use Long.valueOf(String)).\*/

**public** **static** **void** m7() {

String strNumber="123456";

Long valueOf = Long.*valueOf*(strNumber);

System.***out***.println(valueOf);

}

/\* Declare two long variables with values 1123 and 9845, and add them using a method from the Long class.

(Hint: Use Long.sum(long, long)).\*/

**public** **static** **void** add() {

**long** x=1123;

**long** y=9845;

**long** sum = Long.*sum*(x, y);

System.***out***.println(sum);

}

/\*Declare two long variables with values 1122 and 5566, and find the minimum and maximum values using the Long class.

(Hint: Use Long.min(long, long) and Long.max(long, long)).\*/

**public** **static** **void** m8() {

**long** x=1122;

**long** y=5566;

**long** min = Long.*min*(x, y);

System.***out***.println("Minimum Value : "+min);

**long** max = Long.*max*(x, y);

System.***out***.println("Maximum Value : "+max);

}

/\*Declare a long variable with the value 7.

Convert it to binary, octal, and hexadecimal strings using methods from the Long class. (Hint: Use Long.toBinaryString(long), Long.toOctalString(long), and Long.toHexString(long)).\*/

**public** **static** **void** m9() {

**long** l=7;

String binaryString = Long.*toBinaryString*(l);

System.***out***.println("Binary : "+binaryString);

String octalString = Long.*toOctalString*(l);

System.***out***.println("Octal : "+octalString);

String hexString = Long.*toHexString*(l);

System.***out***.println(hexString);

}

**public** **static** **void** main(String[] args) {

*m1*();

*m2*();

*m3*();

*m4*();

*m5*();

*m6*();

*m7*();

*add*();

*m8*();

*m9*();

}

}

**package** com.assignment;

**public** **class** Question\_No\_6 {

/\*

\* Write a program to test how many bytes are used to represent a float value

\* using the BYTES field. (Hint: Use Float.BYTES).

\*/

**public** **static** **void** m1() {

**float** f = 10.6f;

**int** bytes = Float.***BYTES***;

System.***out***.println(bytes);

}

/\*

\* Write a program to find the minimum and maximum values of float using the

\* MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Float.MIN\_VALUE and

\* Float.MAX\_VALUE).

\*/

**public** **static** **void** m2() {

**float** maxValue = Float.***MAX\_VALUE***;

System.***out***.println("Max Value : " + maxValue);

**float** minValue = Float.***MIN\_VALUE***;

System.***out***.println("Min Value : " + minValue);

}

/\*

\* Declare a method-local variable number of type float with some value and

\* convert it to a String using the toString method. (Hint: Use

\* Float.toString(float)).

\*/

**public** **static** **void** m3() {

**float** f = 5.6f;

String string = Float.*toString*(f);

System.***out***.println(string);

}

/\*

\* . Declare a method-local variable strNumber of type String with some value

\* and convert it to a float value using the parseFloat method. (Hint: Use

\* Float.parseFloat(String)).

\*/

**public** **static** **void** m4() {

String strNumber = "1234";

**float** parseFloat = Float.*parseFloat*(strNumber);

System.***out***.println(parseFloat);

}

/\*

\* Declare a method-local variable strNumber of type String with the value

\* "Ab12Cd3" and attempt to convert it to a float value. (Hint: parseFloat

\* method will throw a NumberFormatException).

\*/

**public** **static** **void** m5() {

String strNumber = "Ab12Cd3";

**float** parseFloat = Float.*parseFloat*(strNumber);

System.***out***.println(parseFloat);

}

/\*

\* Declare a method-local variable number of type float with some value and

\* convert it to the corresponding wrapper class using Float.valueOf(). (Hint:

\* Use Float.valueOf(float)).

\*/

**public** **static** **void** m6() {

**float** f = 52.6f;

Float valueOf = Float.*valueOf*(f);

System.***out***.println(valueOf);

}

/\*

\* Declare a method-local variable strNumber of type String with some float

\* value and convert it to the corresponding wrapper class using

\* Float.valueOf(). (Hint: Use Float.valueOf(String)).

\*/

**public** **static** **void** m7() {

String strNumber = "123.12f";

Float valueOf = Float.*valueOf*(strNumber);

System.***out***.println(valueOf);

}

/\*

\* Declare two float variables with values 112.3 and 984.5, and add them using a

\* method from the Float class. (Hint: Use Float.sum(float, float)).

\*/

**public** **static** **void** m8() {

**float** f1 = 112.3f;

**float** f2 = 984.5f;

**float** sum = Float.*sum*(f1, f2);

System.***out***.println("Sum : " + sum);

}

/\*

\* Declare two float variables with values 112.2 and 556.6, and find the minimum

\* and maximum values using the Float class. (Hint: Use Float.min(float, float)

\* and Float.max(float, float)).

\*/

**public** **static** **void** m9() {

**float** f1 = 112.2f;

**float** f2 = 556.6f;

**float** min = Float.*min*(f1, f2);

System.***out***.println("Min : " + min);

**float** max = Float.*max*(f1, f2);

System.***out***.println("Max : " + max);

}

**public** **static** **void** main(String[] args) {

*m1*();

*m2*();

*m3*();

*m4*();

*m5*();

*m6*();

*m7*();

*m8*();

*m9*();

}

}

**package** com.assignment;

**public** **class** Question\_No\_8 {

/\*

\* Initialize a variable of each primitive type with a user-defined value and

\* convert it into String: o First, use the toString method of the corresponding

\* wrapper class. (e.g., Integer.toString()). o Then, use the valueOf method of

\* the String class. (e.g., String.valueOf()).

\*/

**public** **static** **void** conversion() {

**byte** b = 10;

**short** s = 123;

**int** i = 10;

**long** l = 12345;

**float** f = 12.5f;

**double** d = 12.30;

**char** ch = 'A';

**boolean** x = **true**;

String string1 = Integer.*toString*(i);

System.***out***.println(string1);

String string2 = Integer.*toString*(ch);

System.***out***.println(string2);

String string3 = Integer.*toString*(s);

System.***out***.println(string3);

String string4 = Integer.*toString*(b);

System.***out***.println(string4);

String valueOf = String.*valueOf*(b);

System.***out***.println(valueOf);

String valueOf1 = String.*valueOf*(i);

System.***out***.println(valueOf1);

String valueOf2 = String.*valueOf*(f);

System.***out***.println(valueOf2);

}

**public** **static** **void** main(String[] args) {

*conversion*();

}

}

**package** com.assignment;

**public** **class** Question\_No\_9 {

**int** i;

**float** f;

**double** d;

**char** ch;

**boolean** bl;

**long** l;

**byte** b;

**short** s;

**public** **static** **void** main(String[] args) {

Question\_No\_9 check=**new** Question\_No\_9();

System.***out***.println("Default value of int :"+check.i);

System.***out***.println("Default value of float :"+check.f);

System.***out***.println("Default value of double :"+check.d);

System.***out***.println("Default value of char :"+check.ch);

System.***out***.println("Default value of boolean :"+check.bl);

System.***out***.println("Default value of long :"+check.l);

System.***out***.println("Default value of byte :"+check.b);

System.***out***.println("Default value of short :"+check.s);

}

}

**package** com.assignment;

**import** java.util.Scanner;

**public** **class** Question\_No\_10 {

/\*

\* Write a program that accepts two integers and an arithmetic operator (+, -,

\* \*, /) from the command line. Perform the specified arithmetic operation based

\* on the operator provided. (Hint: Use switch-case for operations).

\*/

**public** **static** **void** arithmaticOpeartion() {

Scanner scan = **new** Scanner(System.***in***);

System.***out***.println("Enter the first number :");

**int** num1 = scan.nextInt();

System.***out***.println("Enter the second number :");

**int** num2 = scan.nextInt();

System.***out***.println("Enter the arithmatic operator :");

String choose = scan.next();

**switch** (choose) {

**case** "+":

System.***out***.println("Addition of 2 Numbers : " + (num1 + num2));

**break**;

**case** "-":

System.***out***.println("Subtraction of 2 Numbers : " + (num1 - num2));

**break**;

**case** "\*":

System.***out***.println("Multiplication of 2 Numbers : " + (num1 \* num2));

**break**;

**case** "/":

System.***out***.println("Division of 2 Numbers : " + (num1 / num2));

**break**;

}

}

**public** **static** **void** main(String[] args) {

*arithmaticOpeartion*();

}

}