Module – 6

1. W.A.J.P to Take three numbers from the user and print the greatest number.

```
package automaction;

import java.util.Scanner;

public class Assigment {

	public static void main(String[] args) {
		int num1,num2,num3,total;
		Scanner sc=new Scanner(System.in);
		
		System.out.println("enter first number : ");
		num1=sc.nextInt();
		
		System.out.println("enter second  number : ");
		num2=sc.nextInt();
		
		System.out.println("enter third  number : ");
		num3=sc.nextInt();
		
		int greatest;
	  if (num1 >= num2 && num1 >= num3) {
	      greatest = num1;
	  } else if (num2 >= num1 && num2 >= num3) {
	      greatest = num2;
	  } else {
	      greatest = num3;
	  }
		System.out.println("The greatest number is : 
"+greatest);
	 }

   }


}
```

2. W.A.J.P in Java to display the first 10 natural numbers using while loop.

```
package Assigements;

public class NaturalNumbers {
   public static void main(String[] args) {
          int i=1;
          System.out.println("First 10 Natural Numbers : ");

          while (i<=10)
          {
                System.err.println(i);
                i++;
          }
   }

}
```

3. W.A.J.P to find factorial for Given Number.

```
package Assigements;

import java.util.Scanner;

public class GivenNumber {
   public static void main(String[] args) {
          Scanner sc=new Scanner(System.in);
          System.out.println("Enter a number to find factorial :");
          int num= sc.nextInt();
          long factorial = 1;
          for (int i = 1; i <= num; i++) {
            factorial *= i;
        }
           System.out.println("Factorial of " + num + " is: " +
factorial);

           sc.close();

   }

}
```

4. W.A.J.P to check given number is Prime or not?

```
package Assigements;

import java.util.Scanner;

public class givennumberPrimeornot {
   public static void main(String[] args) {
          int i,num1 = 0;
           boolean isPrime = true;
          Scanner sc=new Scanner(System.in);
          System.out.println("Enter a number to check if prime :  ");
          if (num1 <= 1)
          {
            isPrime = false;
          }
          else {
           for (int i1= 2; i1 <= num1 / 2; i1++) {
                if (num1 % i1 == 0) {
                    isPrime = false;
                    break;

          }

                }
          }
        if (isPrime) {
            System.out.println(num1 + " is a Prime number.");
        } else {
            System.out.println(num1 + " is NOT a Prime number.");
        }
   }
}
```

5. W.A.J.P to check given number is Armstrong or not?

```
package Assigements;

import java.util.Scanner;

public class ArmstrongCheck {
     public static void main(String[] args) {

                   Scanner sc = new Scanner(System.in);
                   System.out.print("Enter a number: ");
                   int number = sc.nextInt();
                   int originalNumber = number;
                   int result = 0, remainder;
                   int n = String.valueOf(number).length();

                   while (number != 0) {
                       remainder = number % 10;
                       result += Math.pow(remainder, n);
                       number /= 10;
                   }

                   if (result == originalNumber)
                       System.out.println(originalNumber + " is an 
Armstrong number.");
                   else
                       System.out.println(originalNumber + " is not an 
Armstrong number.");
               }


   }
```

6. W.A.J.P for create Fibonacci Series.

```
package Assigements;

import java.util.Scanner;

public class FibonacciSeries {
   public static void main(String[] args) {
           Scanner sc = new Scanner(System.in);

          System.out.print("Enter number of terms for Fibonacci series:");
         int n = sc.nextInt();

          int first = 0, second = 1;

          System.out.print("Fibonacci Series up to " + n + " terms: ");
for (int i = 1; i <= n; i++) {
              System.out.print(first + " ");
              int next = first + second;
              first = second;
              second = next;
          }

     }

  }
```

7. W.A.J.P to Print pattern Given Below.

1). 1

12

123

1234

12345

```
package Assigements;

public class Pattern1 {

      public static void main(String[] args) {
              int rows = 5;

              for (int i = 1; i <= rows; i++) {

                  for (int j = i; j < rows; j++) {
                      System.out.print(" ");
                  }


                  for (int k = 1; k <= i; k++) {
                      System.out.print(k);
                  }

                  System.out.println();
              }
      }
}
```

2). 1

12

123

1234

12345

```
package Assigements;

public class BinaryPattern {
       public static void main(String[] args) {
              int rows = 5;
           for (int i = 1; i <= rows; i++) {
                   for (int j = 1; j <= i; j++) {

                       if ((i + j) % 2 == 0)
                           System.out.print("1");
                       else
                           System.out.print("0");
                   }
                   System.out.println();
               }

       }
```

```
3).   1
     2 2
    3 3 3
   4 4 4 4
```

```
package Assigements;

public class RepeatingNumberPattern {
       public static void main(String[] args) {
             int rows = 4;

        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= i; j++) {
                System.out.print(i + " ");
            }
            System.out.println();
        }
       }

}
```

```
4).    *
      * * *
     * * * * *
      * * *
       *
```

```
package Assigements;

public class DesiredPattern {
       public static void main(String[] args) {
            int n = 3; // Height of top half

               // Top half
               for (int i = 1; i <= n; i++) {
                   // Print spaces
                   for (int s = 1; s <= (n - i); s++) {
                       System.out.print(" ");
                   }
                   // Print stars
                   for (int j = 1; j <= (2 * i - 1); j++) {
                       System.out.print("*");
                   }
                   System.out.println();
               }
```

```
                        // Bottom half
                        for (int i = n - 1; i >= 1; i--) {
                            // Print spaces
                            for (int s = 1; s <= (n - i); s++) {
                                System.out.print(" ");
                            }
                            // Print stars
                            for (int j = 1; j <= (2 * i - 1); j++) {
                                System.out.print("*");
                            }
                            System.out.println();
                        }
                 }

          }
```

8. WAP to compute the sum of the first 100 prime numbers.

```
package Assigements;

public class SumFirst100Primes {

   public static void main(String[] args) {

          int count = 0, num = 1, sum = 0;

        while (count < 100) {
            if (isPrime(num)) {
                sum += num;
                count++;
            }
            num++;
        }

        System.out.println("Sum of first 100 prime numbers is: " + sum);
    }

    public static boolean isPrime(int n) {
        if (n <= 1) return false;
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) return false;
        }
        return true;
   }
}
```

9. WAP to sum values of an array.

```
package Assigements;

public class SumArrayValues {
   public static void main(String[] args) {
          int[] numbers = {10, 20, 30, 40, 50}; // You can change or
take input from user
        int sum = 0;
        for (int num : numbers) {
            sum += num;
        }

        System.out.println("Sum of array values: " + sum);

    }

}
```

10. WAP to calculate the average value of array elements.

```
package Assigements;

public class Averagearray {
    public static void main(String[] args) {
          int[] numbers = {10, 20, 30, 40, 50}; // Example array
        int sum = 0;

        for (int num : numbers) {
            sum += num;
        }

        double average = (double) sum / numbers.length;

        System.out.println("Average value of array elements: " +
average);

    }
}
```

11. WAP to calculate the average value of array elements.

```
package Assigements;

public class AverageArray1 {
public static void main(String[] args) {
   int[] numbers = {15, 25, 35, 45, 55};  // Sample array
    int sum = 0;

    for (int num : numbers) {
        sum += num;
    }

    double average = (double) sum / numbers.length;
    System.out.println("Average value of array elements: " + average);
}
}
```

12. WAP to find the index of an array element.

```
package Assigements;

import java.util.Scanner;

public class findindex {
   public static void main(String[] args) {
          int[] array = {10, 20, 30, 40, 50, 60};

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the element to find: ");
        int element = sc.nextInt();

        boolean found = false;

        for (int i = 0; i < array.length; i++) {
            if (array[i] == element) {
                System.out.println("Element " + element + " found at 
index: " + i);
                found = true;
                break;
            }
        }

   }

}
```

13. WAP to find the maximum and minimum value of an array.

```
package Assigements;

public class MaxMinInArray {
 public static void main(String[] args) {
    int[] array = {25, 12, 89, 5, 77, 33};

     int max = array[0];
     int min = array[0];

     for (int i = 1; i < array.length; i++) {
         if (array[i] > max) {
             max = array[i];
         }

         if (array[i] < min) {
             min = array[i];
         }
     }

     System.out.println("Maximum value in the array: " + max);
     System.out.println("Minimum value in the array: " + min);
 }

}
```

16. WAP to Compare Two String.

```
package Assigements;

import java.util.Scanner;

public class CompareStrings {
      public static void main(String[] args) {
             Scanner sc = new Scanner(System.in);
```

```
        System.out.print("Enter first string: ");
        String str1 = sc.nextLine();

        System.out.print("Enter second string: ");
        String str2 = sc.nextLine();

        // Case-sensitive comparison
        if (str1.equals(str2)) {
            System.out.println("Both strings are equal.");
        } else {
            System.out.println("Strings are not equal.");
        }

        // Optional: Case-insensitive comparison
        if (str1.equalsIgnoreCase(str2)) {
            System.out.println("Both strings are equal (ignoring
case).");
        }

        sc.close();
    }

}
```

17. WAP to concatenate a given string to the end of another string.

```
package Assigements;

import java.util.Scanner;

public class ConCatenateStringsDemo {

    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the first string: ");
        String s1= sc.nextLine();
        System.out.println("Enter the second string: ");
        String s2= sc.nextLine();
        //System.out.println("Enter the three string: ");
        String result= s1+s2;
        System.out.println("Concatenated String: "+result);
        sc.close();
    }

}
```

18. WAP to demonstrate try catch block.

```
package Assigements;

import java.util.Scanner;

public class tryCatchExample {
    public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    try
```

```
      {

        System.out.print("Enter the first number: ");
        int num1 = sc.nextInt();

        System.out.print("Enter the second number: ");
        int num2 = sc.nextInt();


        int result = num1 / num2;

        System.out.println("Result of division: " + result);
      }
      catch (ArithmeticException e)
      {

        System.out.println("error: Cannot divide by zero.");
    }
      catch (Exception e)
      {

        System.out.println("An unexpected error occurred: " +
e.getMessage());
    }
      finally
      {

        System.out.println("Program completed.");
    }

    sc.close();

      }

}
```

19. WAP to demonstrate multiple catch blocks.

```
package Assigements;

import java.util.Scanner;

public class MultipleCatchExample {
   public static void main(String[] args) {
      Scanner sc=new Scanner(System.in);
       try {

         System.out.print("Enter the first number: ");
         int num1 = sc.nextInt();

         System.out.print("Enter the second number: ");
         int num2 = sc.nextInt();

         int result = num1 / num2;
         System.out.println("Result of division: " + result);

         int[] arr = {1, 2, 3};
         System.out.println("Accessing array element at index 5: " +
arr[5]);
```

```
     }
       catch (ArithmeticException e)
       {
         System.out.println(" ArithmeticException: Cannot divide by 
zero.");
     }
      catch (ArrayIndexOutOfBoundsException e)
       {
         System.out.println(" ArrayIndexOutOfBoundsException: Invalid array 
index.");
     }
       catch (Exception e)
       {
         System.out.println(" General Exception: " + e.getMessage());
     }
       finally
       {
         System.out.println("Program finished.");
     }

     sc.close();
}
}
```

20. WAP to create one thread by implementing Runnable interface in Class.

```
package Assigements;

import java.util.Scanner;

class MyRunnable implements Runnable {
    public void run() {

        for (int i = 1; i <= 5; i++) {
            System.out.println("Running thread: " + i);
            try {
                Thread.sleep(500);
            }
            catch (InterruptedException e)
            {
                System.out.println("Thread interrupted.");
            }
        }
    }
}
public class RunnableExample {

      public static void main(String[] args) {
             MyRunnable myRunnable = new MyRunnable();

             Thread thread = new Thread(myRunnable);

        thread.start();
              System.out.println("Main thread is running...");
      }

}
```

21. WAP to create one thread by extending Thread class in another Class.

```
package Assigements;

class MyThread extends Thread {
    public void run() {

        for (int i = 1; i <= 5; i++) {
            System.out.println("Child Thread: " + i);
            try {
                Thread.sleep(500);
            } catch (InterruptedException e) {
                System.out.println("Thread interrupted.");
            }
        }
    }
}
public class ThreadExample {

       public static void main(String[] args) {
             MyThread thread = new MyThread();
              thread.start();
              System.out.println("Main thread is running...");
       }

}
```

22. WAP to iterate through all elements in an array list.

```
package Assigements;

import java.util.ArrayList;
import java.util.Iterator;

public class ArrayListIteration {
       public static void main(String[] args) {
             ArrayList<String> nam=new ArrayList<>();
             nam.add("Khushi");
             nam.add("Pooja");
             nam.add("Divya");
             nam.add("Janvi");
             System.out.println("for-each loop:");
             for(String name :nam) {
                    System.out.println(name);
             }
             System.out.println(" traditional for loop:");
             for(int i=0;i<1;i++)
             {
                    System.out.println(nam.get(i));
             }
```

```
              System.out.println("Iterator:");
              Iterator<String> it= nam.iterator();
              while(it.hasNext())
              {
                     System.out.println(it.next());
              }

       }

}
```

23. WAP to update specific array element by given element.

```
package Assigements;

import java.util.Scanner;

public class UpdateArrayElement {
       public static void main(String[] args) {
              Scanner sc=new Scanner(System.in);
              int[] num = {10,20,30,40,50,60};
              System.out.println("original array:");
              for(int nummber :num) {
                     System.out.println(num + "");
                     System.out.print("Enter the index (0-4) to update: ");
               int index = sc.nextInt();

               if (index < 0 || index >= num.length) {
                   System.out.println("Invalid index. Please enter a value 
between 0 and " + (num.length - 1));
               } else {
                   System.out.print("Enter new element to update at index " 
+ index + ": ");
                   int newValue = sc.nextInt();

                   num[index] = newValue;

                   System.out.println("Updated array:");
                   for (int i : num) {
                       System.out.print(num + " ");
                   }
               }

               sc.close();
           }

       }

}
```

24. WAP to remove the third element from a array list.

```
package Assigements;

import java.util.ArrayList;

public class RemoveThirdElement {
   public static void main(String[] args) {
       ArrayList<String> list =new ArrayList<String>();
       list.add("Apple");
       list.add("Banana");
       list.add("Cherry");
       list.add("Date");
       list.add("Orange");
       System.out.println("Original ArrayList: "+ list);
       if(list.size()>=3) {
             list.remove(2);
             System.out.println("After removing third element: "+list);
       }else {
             System.out.println("ArrayList has less than 3 elements.");
       }
}
}
```

25. WAP to Copy one array into another.

```
package Assigements;

public class CopyArray {
       public static void main(String[] args) {
             int[] originalArray = {10,20,30,40,50,60};
             int[] copiedArray  = new int[originalArray.length];
             for(int i=0;i<originalArray.length;i++)
             {
                    copiedArray[i]=originalArray[i];
             }
             System.out.println("copiedArray");
             for(int num :copiedArray)
             {
                    System.out.println(num + " ");
             }
       }

}
```

26. WAP to reverse an array of integer values.

```
package Assigements;

public class ReverseArray {
       public static void main(String[] args) {
             int[] arr= {10,20,30,40,50,60};
             System.out.println("Original Array: ");
             for(int num : arr) {
```

```
            System.out.println(num + " ");
        }
        int start=0;
        int end= arr.length -1;
        while(start<end)
        {
            int temp=arr[start];
            arr[start]=arr[end];
            arr[end]=temp;
            start++;
            end--;
        }
        System.out.println("reversed Array:");
        for(int num:arr)
        {
            System.out.println(num+" ");
        }
    }

}
```

27. WAP to find the second largest element in an array.

```
package Assigements;

import java.util.Arrays;

public class SecondLargestEasy {
    public static void main(String[] args) {
        int[] arr = {12, 45, 67, 23, 89, 45};

        // Sort the array in ascending order
        Arrays.sort(arr);

        // Find the second largest by checking from the end
        int n = arr.length;
        int largest = arr[n - 1];

        for (int i = n - 2; i >= 0; i--) {
            if (arr[i] != largest) {
                System.out.println("Second largest element is: " + 
arr[i]);
                return;
            }
        }

        System.out.println("All elements are the same. No second 
largest found.");

    }

}
```

28. W.A.J.P. Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.

```java
package Assigements;

abstract class Parent {
    abstract void message();
}
class FirstSubclass extends Parent {
    void message() {
        System.out.println("This is first subclass");
    }
}
class SecondSubclass extends Parent {
    void message() {
        System.out.println("This is second subclass");
    }
}
public class TestAbstract {
   public static void main(String[] args) {
          Parent obj1 = new FirstSubclass();
       Parent obj2 = new SecondSubclass();
       obj1.message();
       obj2.message();
}
}
```

29. W.A.J.P. which will ask the user to enter his/her marks (out of 100). Define a method that will display grades according to the marks entered as below:

Marks	Grade
91-100	AA
81-90	AB
71-80	BB
61-70	BC
51-60	CD
41-50	DD
<=40	Fail

```java
package Assigements;

import java.util.Scanner;

public class GradeCalculator {
    static void displayGrade(int marks)
    {
        if(marks >=91  && marks <=100)
        {
            System.out.println("Grade: AA");
        }
        else if(marks >=81 && marks <=90)
        {
            System.out.println("Grade: AB");
        }
        else if(marks >=71 && marks <=80)
        {
            System.out.println("Grade: BB");
        }
        else if(marks >=61 && marks <=70)
        {
            System.out.println("Grade: BC");
        }
        else if(marks >51 && marks <=60)
        {
            System.out.println("Grade: CD");
        }
        else if(marks >=41 && marks <=50)
        {
            System.out.println("Grade: DD");
        }
        else if(marks <= 40 )
        {
            System.out.println("Grade: FAIL");
        }
        else
        {
            System.out.println("Invalid marks entered. Please enter 
marks between 0 and 100.");
        }

    }
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter your marks (out of 100):");
        int marks =sc.nextInt();
        displayGrade(marks);
        }

}
```

30. W.A.J.P. to create a custom exception if Customer withdraw amount which is greater than account balance then program will show custom exception otherwise amount will deduct from account balance. Account balance is:2000 Enter withdraw amount:2500 Sorry, insufficient balance, you need more 500 Rs.To perform this transaction.

```
package Assigements;

import java.util.Scanner;

class InsufficientBalanceException extends  Exception {
      public InsufficientBalanceException(String message) {
       super(message);
      }
      }

public class BankTransaction {
      public static void main(String[] args) {
            int balance=2000;
            Scanner sc=new Scanner(System.in);

            System.out.println("Enter withdraw amount: ");
            int withdrawAmount = sc.nextInt();

            try {
                  if(withdrawAmount > balance)
                  {
                        int shortage = withdrawAmount - balance;
                        throw new InsufficientBalanceException("Sorry,
insufficient balance, you need more " + shortage + " Rs. to perform this
transaction.");
                  }
                  else {
                        balance -= withdrawAmount;
                        System.out.println("Transaction successful!");
                        System.out.println("Remaining Balance: "+balance
+ "RS>");

                  }
            }catch(InsufficientBalanceException e) {
                  System.out.println(e.getMessage());
            }
            sc.close();
      }

}
```