**ITSFT-406-1502: Programming Concepts**

**Assignment 1 Part 1 MOCK**

**Basic Programming Concepts**

**November 2022**

**Marking Scheme**:

|  |  |
| --- | --- |
| KU1.1 – Programming Concepts | 5 marks |
| KU1.2 – Syntax Review | 5 marks |
| KU1.3 – Flow of Code | 5 marks |
| KU3.1, AA1.1 – Code Structure | 12 marks |
| KU2.1 – IDEs | 5 marks |
| **Total:** | **32 marks** |

**Guidelines** (please read):

* This is a closed-book assignment. You are **not** allowed to use any class notes, books, websites, or any other source unless instructed otherwise. Also, the use of computers during this part of the assignment is **not allowed**.
* Copying, or even attempting to copy from such sources or from those around you will result in the enforcement of the current disciplinary procedures.
* If you have any questions during the assignment, kindly ask the invigilator.
* It is required that you write neatly and legibly, and be clear in your answers. Attempting to be vague on purpose will result in marks being deducted.

**Programming Concepts**

*KU1.1 - Describe what is meant by programming concepts*

**5 marks**

1. For each statement below, write whether it is *true* or *false*

|  |  |
| --- | --- |
| **Statement** | **True or False** |
| A computer program can be a list of instructions written in plain English | F |
| The computer without programs is still a very intelligent machine | F |
| Programs are raw facts, such as numbers, letters, and words | FS |
| Pseudo-code is an informal language used during the design stage | T |
| A variable in programming is a program which can change | F |
| A control structure in programming enables the program to make a decision or repeat code | T |
| The syntax of a programming language is made up by a set of rules which need to be followed | T |
| An IDE is a language which is universal | F |
| A programming paradigm is a way or style of programming | T |
| A programming language can only fall under one specific paradigm | F |

1. For each of the following explain what they are, why they are used and give an example:
2. Variable

**A Variable is a location in memory which values can be temporarily stored in, once a program is restarted, the value is lost until it is set again in the program.**

1. Method

**A method is a set of instructions which may need to be repeated, this allows the same code to be re-used without the need of having duplicate code.**

1. Array

**An array is a collection of variables which are stored in a list, this allows us to refer to each one easier than having 10 variables for 10 numbers, this would allow 1 array of 10 spaces to store 10 numbers.**

**Syntax Review**

*KU1.2 - Review the programming language syntax*

**5 marks**

The following piece of code contains 6 syntax errors. The first error is marked as an example. Find the other 5 errors in order to fill in the table below the code. Kindly note that mistakes made up of pairs such as missing parenthesis, etc…only count as 1 mistake.

1. public class int {

2. public void main static(String []args){

3. int i = 0

4. while(i < 10) {

5. System.out.println(i++); }

6. default {

7. System.out.println(Ready!);

8. }

9. }}}

|  |  |  |
| --- | --- | --- |
| **Error** | **Line Number** | **What’s wrong?** |
| Error A | 1 | The class has an invalid name (a keyword) |
| Error B | 2 | static and main need to be switched |
| Error C | 4 | i is always < 10, as its never being incremented |
| Error D | 6 | Default is not a function which can be called, it is used in the switch statement only |
| Error E | 7 | “Ready!” is not wrapped in “” |
| Error F | 9 | Extra } |

**Flow of Code**

*KU1.3 - Explain the importance of the effects of the flow of code*

**5 marks**

1. Mention 2 different control structures in Java giving a code snippet example of how you would use them.

Control Structure 1: **if/else/else if**

**If(input.equals(“1”){**

**System.out.println(“1”);**

**} else if(input.equals(“2”){**

**System.out.println(“2”);**

**} else {**

**System.out.println(“3”);**

**}**

Control Structure 2: **Loops**

**for(int i = 0; i < 10; i++){**

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

**}**

1. Give an example (description, no need for code) where you can use either an if-statement or a switch statement (an example where both would be appropriate).

**An example which allows you to either use a switch statement or an if statement is a menu screen.**

1. Give an example (description, no need for code) where you can **must** use an if-statement rather than a switch statement.

**When comparing inputs with pre-set values**

1. For each code fragment in the table below, write the output you would expect if the code is run within a valid main method, with the necessary imports. You can assume that these fragments have no syntax errors. The output should be accurate and includes also new lines, if any.

|  |  |
| --- | --- |
| **Code** | **Output** |
| String s = “”;  **for** (int i = 3; i > 0; i--)  {  s += “\*”;  }  System.out.println(s); | \*\*\* |
| int x = 10;  **while** (x > 10)  {  System.out.println(x);  } | N/A |
| int x = 0;  **do**  {  System.out.println(x++);  }  **while** (x < 3); | 1  1  1  1  1  1  etc. |
| **for** (int i = 0; i < 10; i++)  {  if (i < 5)  **continue**;  System.out.println(i);  } | 5  6  7  8  9 |

**Code Structure**

*KU3.1 - Explain the basic structure of a program*

*AA1.1 - Outline the code structure so that anyone else can understand it*

**12 marks**

1. Identify the Java keywords in the following code. For each, explain why it is used:

package mytest;

import java.util.List;

class ListTest {

/// code here

}

|  |  |
| --- | --- |
| **Keyword** | **Used for** |
| package | Package refers to the folder storing the .java files |
| import | Import refers to a pre-defined package being imported to be used in the class |
| class | Class is used to wrap the code of a file |

1. Consider the following method signatures:

public static int addAll(int a, int b, int c)

{/\*Code\*/}

public static void doSomething(ArrayList<String> names)

{/\*Code\*/}

Answer the following questions about the methods:

* What do they return?

addAll: **int**

doSomething: **void**

* How many parameters do they require?

addAll:**3**

doSomething: **1**

* Which of the following calls would result in an error and why?

System.out.println(addAll(1,2,3)); //call 1

doSomething(“Jacob”, “Stella”); //call 2

addAll(3, 3.3, 3.6); //call 3

**Call 2 – This requires an ArrayList of type string, rather than directly passing strings**

**Call 3 – This requires the use of integers not doubles**

1. Fill in the blanks in the code below:

import java.util.**Random**;

import java.util.**Scanner**;

public **class** FunctionCall {

public static void funct1 () {

System.**out**.println("GoodBye");

}

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

Random r = **Random** Random();

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

**int** random = r.nextInt(10);

System.out.print("Guess the number: ");

int guess = s.**readInt**();

if (random == guess)

System.out.println("Well Done");

funct1();

}

}

**IDEs**

*KU2.1 - Explain what an IDE is and how to use it*

**5 marks**

NetBeans is considered to be an IDE. Answer the following questions about IDEs.

1. What is an IDE? Mention 4 useful features

**An IDE is an application which provides the necessary tools and environment to develop an application in a specific programming language.**

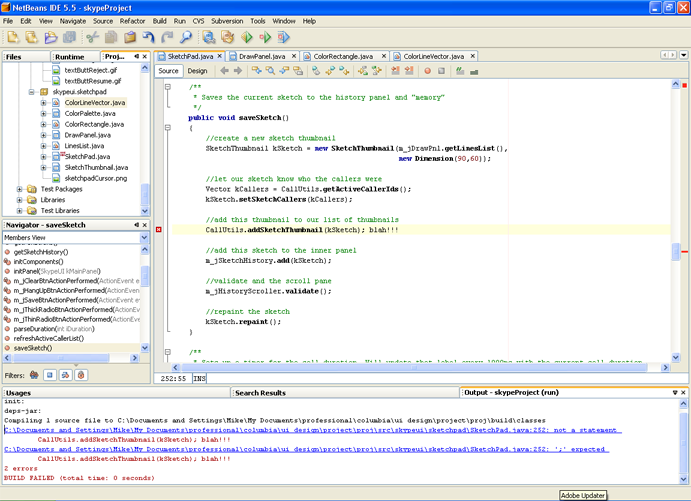
Feature 1: **Code Editor**

Feature 2: **Debugger**

Feature 3: **Project Panel**

Feature 4: **Building**

b) Explain the role of the 3 panels highlighted in the screenshot below:



1

3

2

Panel 1: **This allows the programmer to see what projects they have open**

Panel 2: **The editor allows the programmer to edit the code of the class selected**

Panel 3: **Shows the programmer about any errors while building, or and outputs which have been logged**

*---------------------------------------------End of Mock Assignment Part 1------------------------------------------------*

**ITSFT-406-1502: Programming Concepts**

**Assignment 1 Part 2 MOCK**

**Basic Programming Concepts**

**November 2022**

**Marking Scheme**:

|  |  |
| --- | --- |
| KU2.2, KU3.2, AA3.1 - Program | 17 marks |
| **Total:** | **17 marks** |

**Guidelines** (please read):

* This is a closed-book assignment. You are **not** allowed to use any class notes, books, websites, or any other source unless instructed otherwise.
* Copying, or even attempting to copy from such sources or from those around you will result in the enforcement of the current disciplinary procedures.
* If you have any questions during the assignment, kindly ask the invigilator.
* Upload file on VLE(in assignment) and send a copy to your lecturer on teams chat just in case

**Write a program given its specification, using appropriate data types**

*KU2.2 - Recognize and use the main views of an IDE*

*KU3.2 - Distinguish the appropriateness of different types of variables used for different contexts*

*AA3.1 - Interpret a program specification to write a small piece of code*

**17 marks**

Create a new NetBeans project, and implement the 3 programs that produce the following output. (Note: bold indicates user input).

Program 1:

Enter a number: **5**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

Program 2: (make use of an ArrayList to store the names, and create methods)

Enter a list names of your friends (type ‘stop’ to end input)

Name: **John**

Name: **Patrick**

Name: **Melissa**

Name: **Frederick**

Name: **stop**

You entered 5 names. The longest is Frederick.

Program 3: (make sure you create methods)

Enter 10 numbers:

**5**

**7**

**8**

**4**

**9**

**9**

**6**

**6**

**1**

**123**

Sum: 178

Average: 17.8

Largest: 123

Smallest: 1

*---------------------------------------------End of Mock Assignment Part 2------------------------------------------------*