CONVERSIONS NAME: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* Understand binary, hexadecimal & decimal bases
* Input (numeric & string)
* Looping
* Branching
* Print formatting

|  |  |  |  |
| --- | --- | --- | --- |
| **OBJECTIVE** | **DESCRIPTION** | **RESULTS EXPECTED** | **POINTS\*** |
| **0. Pseudocode** | Pseudocode provided is clear, easy to follow and allows the program to follow program logic |  | **/20** |
| **1. Accuracy** | Menu provided for user is clear, includes all commands,  & is easy to read |  | **/10** |
|  | Incorrect selection provides an error |  | **/5** |
|  | a. input: 32  base: 16 | The value 32 in base 16 is 50 in decimal | **/10** |
|  | a. input: 11001100  base: 2 | The value 11001100 in base 2 is 204 in decimal | **/10** |
|  | a. input: 95  base 10 | The value 95 in base 10 is 95 in decimal | **/10** |
|  | b. input: 255  base 2 | The decimal value 255 in 11111111 in base 2 | **/10** |
|  | b. input: 171  base 16 | The decimal value 171 is AB in base 16 | **/10** |
|  | b. input: 52  base 8 | The decimal value 52 is 64 in base 8 | **/10** |
|  | b. input: 171  base 16 | The decimal value 171 is AB in base 16 | **/10** |
|  | b. input: AA  base 2 | Input error | **/10** |
| **2. Style** | Code is well organized  & easy to read | * Related code is put together * Indentions are properly used to improve readability * Variable names are meaningful | **/10** |
|  | Code is well commented | * Heading information includes Name, Date & Program name * Brief explanations before blocks of related code | **/10** |
| **Bonus Feature(s)** | Loop until ‘q’ is entered |  | **+10** |
|  | Code will convert from any base to any base |  | **+20** |
| **TOTAL** | ADDITIONAL COMMENTS: |  | **/135** |

**POINTS\***

Points assigned are partially subjective, but you can expect points assigned as follows :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 1 | 2 - 3 | 4 | 5 (5 point scale) |
| 0 | 1-5 | 6 – 7 | 8-9 | 10 (10 point scale) |
| 0 | 1-10 | 12-14 | 15-19 | 20 (20 point scale) |
| No evidence of meeting this requirement | Evidence of requirement, but working at a substandard level  (code is present, but does not compile, for example) | Evidence of requirement, partially working  (code is present, some values work, but other values cause an abend for example) | Requirement is predominantly met, but has some small issues  (code is present and runs, but output is not completely correct, for example) | Requirement is fully met and functioning |